Stratified Sampling Frame in GIS: A Computer Code for Latin Hypercube Sampling (LHS) Technique in Environmental Modelling

Festus K. Ng'eno

This thesis has been submitted to the **Department of Environmental and Biosystems Engineering** of the **University of Nairobi** in partial fulfillment of the requirements for the award of **Master of Science** in **Environmental and Biosystems Engineering.**

UNIVERSITY OF NAIROBI

August 2010

Declaration

I, Festus K. Ng'eno, hereby declare that this MSc. thesis is my original work. To the best of my knowledge, the work presented here has not been submitted for a degree programme in any university.

Festus K. Ng'eno

19|08|10

Date

This thesis has been submitted for examination with our approval as university supervisors.

Dr. Christian Omuto

19/08/2010

Date

Prof. Elijah K. Biamah

Department of Environmental and Systems Engineering
University of Nairobi
P. O. BOX 30197
NAIROBI

19/08/2010

Date

Dedication

This work is dedicated to the memory of…

my father, J. K. Koskei, who guided me to the right path;

and who left when it had just begun …

.........to the two inspiring men towards my study......

my dear supervisor, Dr. C. Omuto, for his visions in the use of ArcView software in environmental modelling;

my dear co-supervisor, Prof. E.K. Biamah, who always wanted me to have the best.

.... to the two most important women in my life......

my mother, G. Koskei, for her love, prayers, pains, troubles and guidance since my childhood and

my loving wife, Dr. C. Ndeta Ng'eno, for her constant long calls which gave a sense of love and for every good thing a wife would offer to a husband.

...... and my ever loving and caring extended family…

Acknowledgements

First and foremost, I would like to give glory to our God Almighty for His Grace and help in all my endeavors and for bringing me this far with my education. I would like to express my gratitude to my supervisor Dr. C. Omuto for his guidance throughout this research; his comments and constructive criticisms, which greatly enhanced this thesis. Further, he is thanked for providing advice on the use of ArcView software in environmental modelling and helping me to gain dynamic research skills to complete this thesis.

More thanks go to the academic and non-academic staff of the School of Engineering. People who stay and work in the shadows and whose praises are yet to be sang to a high crescendo with special thanks to the wonderful Environmental & Biosystems Engineering staff led by my Co-supervisor and Chairman, Prof. E.K. Biamah, for advice on numerous technical issues about GIS modelling and for providing computing support.

What else can the user say to his friends and brothers who during the three years of research study have shared the tears, sorrows and joy with him? People who in one way or the other have been instrumental in making my study a memorable and life-changing experience. Would "thank you" be enough for providing valuable assistance and a research environment that was productive?

Back to my village, I want to thank my beloved mother for her patience, perseverance and understanding throughout this period. I fondly remember you. I am grateful to my brothers and sisters. God bless you all. And above all, lots of love goes to my dear wife, Dr. C. Ndeta Ng'eno – many thanks for your love and support.

Finally, no words can describe my debt to my sponsors (Board of Post Graduate Studies – University of Nairobi) and my family for their moral support during every single moment of all my graduate studies.

It has been nice knowing you all; and I say to you all, God Bless You.

TABLE OF CONTENTS

List of Figures

List of Tables

List of Appendices

List of Abbreviations

AVSWAT	–	ArcView Soil and Watershed Assessment Tool
CSA	–	Critical Source Code
dBASE	–	Data Base
DEM	–	Digital Elevation Model
GeoWEPP	–	Geospatial Water Erosion Prediction Project
GIS	–	Geo-Information System
GUI	–	Graphical User Interface
INFO	–	Information
LHS	–	Latin Hypercube Sampling
NDVI	-	Normalized Difference Vegetation Index
ODB	–	Object Data Base
OOP	-	Object Oriented Programming
SWAT	–	Soil and Water Assessment Tool
SWAT-CUP	-	SWAT Calibration and Uncertainty Procedures
USLE	–	Universal Soil Loss Equation
WEPP	–	Water Erosion Prediction Project

Abstract

Field sampling is a widely applied data collection method in environmental engineering. Sampling, which is supposed to provide adequate and unbiased representation of a study interest, is often faced with challenges such as choice of the minimum cost-effective sampling size, accurate location of sampling sites and statistical justification of both location of the sites and accuracy of the sampling process. Although there are many statistical sampling protocols in the literature, they largely emphasize on accuracy tests for unbiased sampling size. There is still a lack of statistical routines for unbiased sample locations. This has led to a lot of errors in spatial modeling of many environmental processes. In spite of the spatial modeling opportunities in GIS, there is currently no implementation of efficient statistical sampling frameworks within the GIS to improve sampling accuracy during primary data collection. This study developed a computer program for implementing a Latin Hypercube Sampling (LHS) protocol within a GIS environment.

LHS is a robust and efficient statistical framework that has been widely used to guide the determination of sample sizes given a set of constraints. When combined with GIS software such as ArcView, LHS can improve sample size selection and subsequent unbiased strategic location of the samples in the landscape. This study developed a computer code for implementing LHS in ArcView. ArcView scripts were written in avenue language for various applications in environmental engineering. The scripts were then integrated into one unit to produce the LHS extension in ArcView GIS.

The LHS extension developed has several features which can help environmental engineers and/or other field scientists to carry out their studies. The features include: random point sampling and systematic point sampling. Random point sampling feature creates a shapefile of randomly placed points within a polygon or selected polygons in an active theme. It can also be used to randomize specific features in an active theme. For example, it can be used to randomly select a number of bounded areas or zones in a given study area as opposed to random placement of points in such areas.

Systematic sampling feature creates a systematic network of points within the selected polygon(s) or graphic(s). This feature relies on arranging the target population according to some ordering scheme and then selecting elements at regular intervals through that order.

In this thesis, two objectives which formed the integral part of the study were met; that is, the development of a computer code for Latin Hypercube Sampling (LHS) method using avenue computer language, and the development of an add-on program in ArcView GIS for implementing the LHS code. LHS in ArcView is easy to implement since it uses GUI's controls which are user friendly.

For further success, improvements and wide applications of the LHS extension in ArcView developed in this study, the following recommendations were made: more features in the LHS extension be developed and further discussed in detail with relevant engineering applications and that LHS be tested with data from other parts of the world.

1: INTRODUCTION

1.1 Background

Many studies in environmental engineering involve application of models which try to simulate the functioning of environmental processes. The motivation in developing these models is often to explain the complex behavior in environmental systems or improve understanding of a given system. They are sometimes extrapolated in time/space in order to predict future environmental conditions or to compare predicted behaviour to the observed processes or phenomena (Skidmore, 2002). In general, models and modeling are part and parcel of environmental engineering.

Although models have important applications in engineering, their development and application are not always useful unless the data used in their development and validation has been gathered with acceptable accuracy. One of the main challenges in data collection is unbiased representation of all the variability and calibration ranges for the intended models. Especially for large areas, data collection for environmental studies is very cumbersome, expensive and takes too long for every location of interest. In many cases, the samples are always few, emanating from easy-to-access locations, and are unrepresentative of landscape variability. Consequently, the models using such datasets give erroneous results and can potentially mislead environmental decisions. In other applications, samples in data collection are too many and take too long to obtain so that by the time the last sample is collected many environmental changes have occurred in the first sample. Similarly, the models using such datasets result in wrong representation of the environmental conditions. There is need therefore, for proper guidance on data collection to target sample sizes and location of their sites in the landscape.

The standard practice to circumvent the challenges in data collection is through taking representative samples that are assumed to have similar statistical characteristics as the entire study area (Kottegoda and Rosso, 1998). Literature is replete with statistical techniques for determining the required minimum sample-size to meet such a criterion. Examples such as random sampling, stratified random sampling and Latin hypercube sampling have been extensively used in many studies to achieve this goal (Stein, 1987; Minasny and McBratney, 2006). In environmental studies, not only is the choice of minimum sampling-size important but also the strategic location of the sites in the study area of interest. The reasons are two-fold: to capture landscape variability and to cover the landscape sufficiently for reasons of spatial extrapolation (Nielsen and Wendroth, 2003). This study anticipated combining site selection criterion in the sample-size determination for strategic location of a minimum number of sample sizes necessary to study any environmental processes in the landscape.

1.2 Problem statement and justification

The choice of sample-size is an important research aspect in environmental studies since it determines the accuracy of relating statistical characteristics of the samples to that of the entire population. Besides the determination of sample-size, unbiased allocation of samples to representative parts of a study area is also important. Sampling protocol is the procedure for collecting data for environmental analysis by providing sufficient coverage of the calibration ranges of the environmental models used and also endeavouring to visit all parts of a study area. Efficient sampling protocol guarantees accurate modeling while minimizing the cost, drudgery and time for sampling. Many methods have been proposed in the literature for sampling protocols, some of which include simple random sampling, stratified random sampling, variance quadrature tree, Latin hypercube sampling, among many others.

The problems of sampling protocols in environmental engineering are two-fold: first, the choice of minimum sample-size to guarantee accurate modeling and second allocation of the samples in space for accurate extrapolation. Minimum sample size is important in reducing cost and time for sampling while strategic location of samples in the landscape ensures sufficient coverage of both geographic and feature spaces needed to capture all calibration ranges in the environmental models (Omuto and Vargas, 2009). Most existing sampling protocols hardly meet these requirements. Although there are some protocols with good performance (such as Latin Hypercube Sampling, LHS) for determination of minimum sample sizes, they do not consider sample location in their designs (Minansy and McBratney, 2007). Such protocols can be re-modified to include sample location in the geographic space.

There are opportunities with GIS for manipulating the geographic data in a way that can improve the performance of existing sampling protocols. If given the sampling points, GIS can play around with placement of samples to circumvent accessibility problems (Hengl et al., 2007). Although currently GIS does not have the capability of good sampling protocol (such as in LHS), it has the promise of improving the performance of such protocols. This study produced a sampling protocol within ArcView by combining LHS and GIS capabilities. The study incorporated LHS in ArcView GIS to allow for the optimal determination of sample-sizes and the location of the samples in a study area.

The relevance of this study and its engineering application was exemplified in the importance of sampling in environmental studies. Higher costs are incurred by not following unbiased and representative sampling in environmental studies. This is due to the high variability in natural processes both in space and time. Physical sampling has proven to be an expensive, cumbersome and time consuming exercise. Thus, there is a growing need for a statistically sound sampling frame to guide placement of sample points and choice of minimum number of samples. Although there are statistical routines to guide choice of sample points and those for placement of sample points, there is still a lack of efficient sampling framework within GIS to support these developments. In spite of good sampling techniques being available, they are not well matriculated into the GIS system to support engineers in choosing sample locations statistically and improving on spatial modeling and eventual accurate decision support. This study tried to solve the foregoing problem by using the Latin Hypercube Sampling (LHS) in a GIS environment.

1.3 Objectives of study

1.3.1 Overall objective

To develop a computer program for implementing Latin Hypercube Sampling (LHS) in ArcView GIS for guiding field surveys in environmental engineering studies.

1.3.2 Specific objectives

a) To develop scripts for Latin Hypercube Sampling (LHS) method using avenue computer language.

b) To use scripts developed in (a) above for producing an add-on program in ArcView GIS.

2: REVIEW OF LITERATURE

2.1 Definition of sampling

In many disciplines, there is often the need to describe the characteristics of large entities, such as the air quality in a region, the prevalence of smoking in the general population, or the output from a production line of a pharmaceutical company. Due to practical considerations, it is impossible to assay the entire atmosphere, interview every person in the nation, or test every pill (Kelsey et al., 1986). Opinion pollsters use sampling to gauge political allegiances or preferences for brands of commercial products, whereas water quality engineers employed by public health departments will take samples of water to make sure it is fit to drink. The process of drawing conclusions about the larger entity based on the information contained in a sample is known as statistical inference (Pagano and Gauvreau, 2000).

Sampling is the process of obtaining information from selected parts of an entity, with the aim of making general statements that apply to the entity as a whole, or an identifiable part of it (Kelsey et al., 1986). It is the act, process, or technique of selecting a suitable sample, or a representative part of a population for the purpose of determining parameters or characteristics of the whole population (Webster, 1985). A sample is a finite part of a statistical population whose properties are studied to gain information about the whole population (Kelsey et al., 1986).

2.1.1 Sampling protocols

Sampling protocol is the procedure used to select units from the study population to be measured. The goal of the sampling protocol is to select units that are representative of the study population with respect to the attribute(s) of interest. The sampling protocol deals with how and when the units are selected. It requires a plan, target population and sample size to be decided a priori (Kelsey et al., 1986). The population is defined in keeping with the objectives of the study (http://www.statpac.com/surveys/index.htm) – (access date, 6th September, 2009). Sometimes, the entire population will be sufficiently small, and the researcher can include the entire population in the study (Pagano and Gauvreau, 2000). This type of research is called a census study because data is gathered on every member of the population (Pagano and Gauvreau, 2000). Usually, the population is too large for the researcher to attempt to survey all of its members. A small, but carefully chosen sample can be used to represent the population. The sample reflects the characteristics of the population from which it is drawn (Webster, 1985).

Sampling protocol, therefore, can be regarded as a procedure for collecting data for further (say environmental) analysis by providing sufficient coverage of the calibration ranges of the environmental models used and also endeavouring to visit all parts of a study area (Kelsey et al., 1986). Efficient sampling protocol guarantees accurate modeling while minimizing the cost, drudgery and time for sampling. Many methods have been proposed in the literature for sampling protocols, some of which include simple random sampling, stratified random sampling, variance quadrature tree, Latin hypercube sampling, among many others.

These sampling protocols are classified as either *probability* or *non-probability* (Kelsey et al., 1986). In probability samples, each member of the population has a known non-zero probability of being selected. Probability methods include random sampling, systematic sampling and stratified sampling. In non-probability sampling, members are selected from the population in some non-random manner. These include convenience sampling, judgment sampling, quota sampling and snowball sampling. The advantage of probability sampling is that sampling error can be calculated. Sampling error is the degree to which the outcome from a sample might differ from that of a population. When inferring to the population, results are reported plus or minus the sampling error. In non-probability sampling, the degree to which the sample differs from the population remains unknown.

2.1.2 Importance of sampling in environmental engineering

There are several advantages/importance of using sampling rather than conducting measurements on an entire population. An important advantage is the considerable savings gained in terms of time and money that can result from collecting information from a much smaller population (Kelsey et al., 1986). When sampling individuals, the reduced number of subjects that need to be contacted may allow more resources to be devoted to finding and persuading non-respondents to participate. The information collected using sampling is often more accurate, as greater effort can be expended on the training of interviewers, more sophisticated and expensive measurement devices can be used, repeated measurements can be taken and more detailed questions can be posed (Salant and Dillman, 1994).

In sampling, the researcher only studies a representative part of the population rather than the whole population (Kelsey et al., 1986), i.e. sampling provides the researcher with the much needed information quickly. Sampling helps in studies of large populations which are cumbersome to carry out and therefore, studying a representative sample overcomes the challenge (Kelsey et al., 1986). Sampling helps in carrying out surveys or studies in areas which are inaccessible or partly accessible, for example, in remote areas with difficult terrain or areas with security problems (Salant and Dillman, 1994).

2.2 Definition of stratified sampling

In statistics, stratified sampling is a method of sampling from a population (Bartlett et al., 2001). When sub-populations vary considerably, it is advantageous to sample each sub-population (stratum) independently. Stratification is the process of grouping members of the population into relatively homogeneous subgroups before sampling (Bartlett et al., 2001). The strata should be mutually exclusive (i.e. every element in the population must be assigned to only one stratum). The strata should also be collectively exhaustive (i.e. no population element can be excluded). Then random or systematic sampling is applied within each stratum. This often improves the representativeness of the sample by reducing the sampling error. It can produce a weighted mean that has less variability than the arithmetic mean of a simple random sample of the population (http://www.coventry.ac.uk/ec/~nhunt/meths/strati.html) – (access date, 6th September, 2009).

A stratified sampling approach is most effective when three conditions are met;

a) Variability within strata is minimized.
b) Variability between strata is maximized.
c) The variables upon which the population is stratified are strongly correlated with the desired dependent variable.

2.2.1 Importance of stratified sampling in environmental engineering

There are several potential benefits/importance of using stratified sampling protocol in environmental engineering. First, dividing the population into distinct, independent strata can enable researchers to draw inferences about specific subgroups that may be lost in a more generalized random sample (Bartlett et al., 2001).

Second, utilizing a stratified sampling method can lead to efficient statistical estimates (provided that strata are selected based upon relevance to the criterion in question, instead of availability of the samples). It is important to note that even if a stratified sampling approach does not lead to increased statistical efficiency; such a tactic will not result in less efficiency than would simple random sampling, provided that each stratum is proportional to the group's size in the population (Salant and Dillman, 1994).

Third, it is sometimes the case that data are more readily available for individual, pre-existing strata within a population than for the overall population. In such cases, using a stratified sampling approach may be more convenient than aggregating data across groups (Kelsey et al., 1986).

Finally, since each stratum is treated as an independent population, different sampling approaches can be applied to different strata, potentially enabling researchers to use the approach best suited (or most cost-effective) for each identified subgroup within the population (Kelsey et al., 1986).

There are, however, some potential drawbacks to using stratified sampling. First, identifying strata and implementing such an approach can increase the cost and complexity of sample selection, as well as leading to increased complexity of population estimates (Kelsey et al., 1986). Second, when examining multiple criteria, stratifying variables may be related to some, but not to others, further complicating the design, and potentially reducing the utility of the strata (Salant and Dillman, 1994). Finally, in some cases (such as designs with a large number of strata, or those with a specified minimum sample size per group), stratified sampling can potentially require a larger sample than would other methods - although in most cases, the required sample size would be no larger than would be required for simple random sampling (Bartlett et al., 2001).

2.3 Stratified sampling frame in GIS

2.3.1 Definition of GIS

GIS is a science, a technology, discipline and an applied-problem solving methodology (Longley et al., 2005). It is concerned with the description, explanation and prediction of patterns and process at geographic scales. GIS describes any information system that integrates stores, edits, analyzes, shares and displays geographic information. It is a special kind of information system, distinguished from other information systems by the fact that it deals with spatially referenced data. GIS is a computer based information system to input, retrieve, process, analyze and produce geographically referenced data in order to support decision making, planning and management of natural resources and environment (Longley et al., 2005). The components of GIS include: network, hardware, software, data, management (organization) procedures and people (Longley et al., 2005).

GIS is widely used in natural resources management, facilities planning, transportation routing and logistics, and geo-demographic analysis (Longley et al., 2001). In institutional research, GIS has been used to map the locations of current students or applicants, visualize demographic change in communities, add geo-demographics to enrollment models and for exploration of locations for new campuses (see, for example; Harrington, 2000; Mailloux and Blough, 2000; Acker and Brown, 2001;Pottle, 2001; Wu and Zhou, 2001).

2.3.2 ArcView GIS

ArcView is Geographic Information System (GIS) software for visualizing, managing, creating and analyzing geographic data. ArcView GIS provides tools for working with maps, database tables, charts and graphics all at once. It is able to do a wide array of mapping and analysis functions. The importances of ArcView GIS are:

a) It can produce maps and interact with the data by generating reports, charts, printing and embedding the maps in other documents and applications.
b) It saves time using map templates to create a consistent style in the maps.
c) It helps in building process models, scripts and workflows to visualize and analyze data.

2.4 Latin Hypercube Sampling (LHS)

Latin Hypercube Sampling (LHS) is a form of stratified sampling that can be applied to multiple variables (McKay et al., 1979). The method was commonly used to reduce the number of runs necessary for a Monte Carlo simulation to achieve a reasonably accurate random distribution. LHS can be incorporated into an existing Monte Carlo model fairly easily and work with variables following any analytical probability distribution. LHS is a method for stratifying each univariate margin simultaneously. McKay et al., (1979) introduced LHS for reducing the variance of Monte Carlo simulations.

The statistical method of LHS was developed to generate a distribution of plausible collections of parameter values from a multidimensional distribution. The sampling method is often applied in uncertainty analysis. The technique was first described by McKay et al., (1979). It was further elaborated by Iman and Conover (1981). In the context of statistical sampling, a square grid containing sample positions is a Latin square if (and only if) there is only one sample in each row and each column. A Latin hypercube is the generalization of this concept to an arbitrary number of dimensions, whereby each sample is the only one in each axis-aligned hyperplane containing it.

Monte Carlo simulations provide statistical answers to problems by performing many calculations with randomized variables and analyzing the trends in the output data. The concept behind LHS is not overly complex. Variables are sampled using an even sampling method and then randomly combined sets of those variables are used for one calculation of the target function.

For example, when sampling a function of ***N*** variables, the range of each variable is divided into ***M*** equally probable intervals. ***M*** sample points are then placed to satisfy the Latin hypercube requirements. Note that this forces the number of divisions, ***M***, to be equal for each variable. Also note that this sampling scheme does not require more samples for more dimensions (variables); this independence is one of the main advantages of this sampling scheme. Another advantage is that random samples can be taken one at a time, remembering which samples were taken so far. The process is quick, simple and easy to implement (Iman and Conover, 1981). It helps to ensure that the Monte Carlo simulation is run over the entire length of the variable distributions, taking even unlikely extremes into account as the researcher/field scientist would desire.

2.5 Use of ArcView extensions in environmental engineering/sampling

2.5.1 ArcView extension for site-specific soil and water conservation

2.5.1.1 Soil erosion assessment

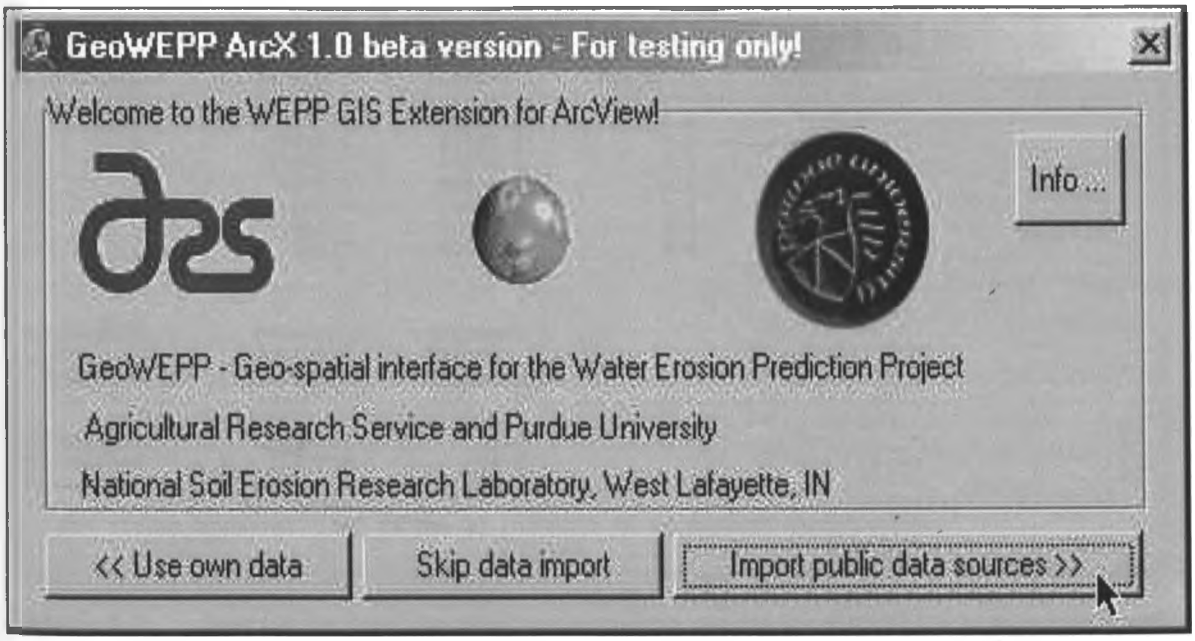
There is a long history of developing erosion models for soil and water conservation in U.S. research. The most known and applied approach for estimating long-term average annual soil loss is the Universal Soil Loss Equation (USLE) (Wischmeier and Smith 1978) and the Revised Universal Soil Loss Equation (RUSLE) (Renard et al., 1997). Both are simple empirical equations based on factors representing the main processes of soil erosion. USLE and RUSLE have proven to be practical, accessible prediction tools and were therefore implemented in the U.S. soil and water conservation legislation. However, these model approaches have been used and misused widely at various scales worldwide (Wischmeier, 1976).

2.5.1.2 Water Erosion Prediction Project (WEPP)

In contrast to the empirical model approaches, efforts in erosion process research in the U.S. lead to the development of the process-based hillslope soil erosion model WEPP (Flanagan and Nearing, 1995). WEPP simulates climate, infiltration, water balance, plant growth and residue decomposition, tillage and consolidation to predict surface runoff, soil loss, deposition and sediment delivery over a range of time scales, including individual storm events, monthly totals, yearly totals or an average annual value based on data for several decades. The WEPP model is a continuous distributed-parameter soil erosion assessment tool that can be applied to representative hillslopes and a channel network at small watershed scales (Ascough et al., 1997). A comparison of the performance of WEPP with other state-of-the-art erosion models using common data sets showed that data quality is an important consideration and primarily process-based models not requiring calibration have a competitive edge to those in need of calibration (Favis-Mortlock, 1998).

2.5.1.3 GIS interface for the WEPP model

GIS in model linkages are dominantly used for data preprocessing and visualization of available data sources as well as the handling of data to apply environmental assessment models. A GIS-driven graphical user interface is a user-friendly approach to combine the decision-support of an environmental prediction model and the spatial capabilities of a GIS for practical assessment purposes (Renschler et al., 2000). A useful and successful implementation of an environmental model assessment approach requires the use of widely available data sets and the preparation of model input parameters to allow reliable model predictions. The prototype of a GIS-based interface - the Geospatial Interface for WEPP (GeoWEPP) – it is an interface for using WEPP through a wizard in ArcView 3.2 for Windows 98, 2000 and NT. The currently released testing version of GeoWEPP ArcX 1.0 beta is an ArcView project/extension that starts with a user-friendly wizard (Fig. 2.1).


GeoWEPP ArcX 1.0 beta version - For testing only!

Welcome to the WEPP GIS Extension for ArcView!

GeoWEPP - Geo-spatial interface for the Water Erosion Prediction Project

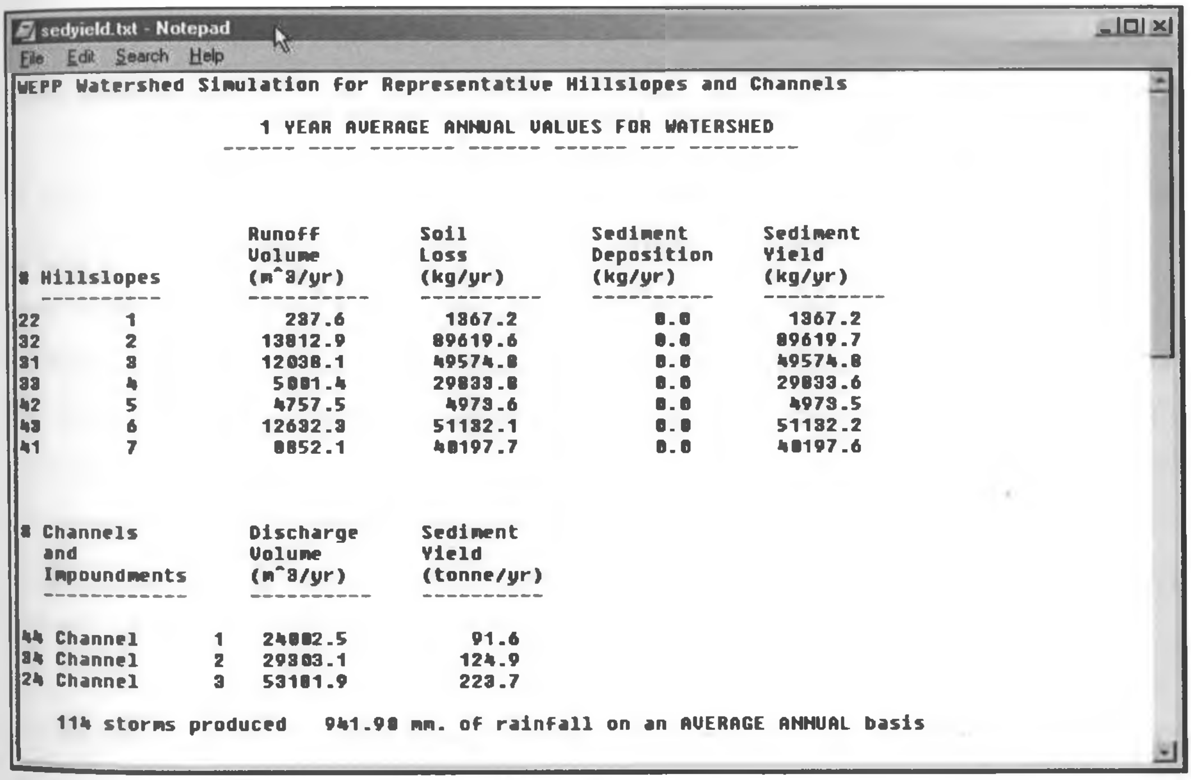
Agricultural Research Service and Purdue University

National Soil Erosion Research Laboratory, West Lafayette, IN

Figure 2.1: Opening screen of ArcView-based GeoWEPP wizard.

2.5.1.4 Assessment results and discussion

The WEPP model creates numerous outputs to its model components, including Climate Simulation, Subsurface Hydrology, Water Balance, Plant Growth, Residue Decomposition and Management, Overland Flow Hydraulics, Hillslope Erosion Component, Channel Flow Hydraulics and Channel Erosion Surface. The current wizard allows the researcher to visualize only a small portion of the WEPP model output of runoff, soil loss, sediment deposition and sediment yield from hillslopes and channel segments. The average annual simulation results for the WEPP Watershed Method are displayed as text file (Fig. 2.2) and visualized as a map.



```
WEPP Watershed Simulation for Representative Hillslopes and Channels

                  1 YEAR AVERAGE ANNUAL VALUES FOR WATERSHED
                  ------ ---- ------- ------ ------ --- ---------


                   Runoff        Soil          Sediment      Sediment
                   Volume        Loss          Deposition    Yield
# Hillslopes       (m^3/yr)      (kg/yr)       (kg/yr)       (kg/yr)
  ----------       ----------    ----------    ----------    ----------
22        1          237.6        1367.2           0.0        1367.2
32        2        13812.9       89619.6           0.0       89619.7
31        3        12038.1       49574.8           0.0       49574.8
33        4         5001.4       29833.8           0.0       29833.6
42        5         4757.5        4973.6           0.0        4973.5
43        6        12632.3       51132.1           0.0       51132.2
41        7         8852.1       40197.7           0.0       40197.6


# Channels         Discharge     Sediment
  and              Volume        Yield
  Impoundments     (m^3/yr)      (tonne/yr)
  ------------     ----------    ----------

44 Channel     1   24002.5          91.6
34 Channel     2   29303.1         124.9
24 Channel     3   53101.9         223.7

  114 storms produced   941.90 mm. of rainfall on an AVERAGE ANNUAL basis
```

Figure 2.2: Text file for average annual simulation results for the WEPP watershed method.

2.5.2 AVSWAT: An ArcView GIS extension as tool for the watershed control

ArcView GIS, extended and integrated with a hydrologic non-point pollution model (SWAT), provides a comprehensive watershed assessment tool (AVSWAT) designed to assist water resource managers (Arnold et al., 1993). AVSWAT improves the efficiency of analysis for non-point and point pollution assessment and control on watershed scale. The watershed modeling framework is delineated starting from the digital description of the landscape (DEM, land use and soil data sets) using ArcView Spatial Analyst with geomorphological assessment procedures and can integrate nationwide public domain databases as well as operate on user provided input data. AVSWAT is a user friendly, unique and single modeling environment based on several user interface tools developed using Dialog Designer extension and is able to run on PC as well as on UNIX platforms.

In the assessment and control of pollutants released from agricultural fields as well as urban areas together with their pathways towards and within the stream network, the river watershed takes its place as a fundamental landscape unit upon which research, analysis, design and planning are based. This is done side by side with consideration of the hydrologic cycle, erosion and delivery of sediments and agricultural management practices. As the practical consideration and public concern on water quality there is an increasing need of globally applicable assessment tools which at once identifies and indicates the spatial boundaries and geomorphic characteristics of a hydrographic basin and its sub-units together with their hydrologic parameters.

Moreover, it defines the climate and other parameters inputs for hydrologic simulation models that have revealed their effectiveness to finally focus the areas of high rate of pollutant release, waters that do not meet state water quality standards and evaluate the capabilities of best available alternative control measures. To address this needs, the presented ArcView extension, brings to the user a set of several tools working in a sequential order starting from the delineation and codification of watershed based upon the topography and ending with the analysis and calibration of the hydrologic simulations of SWAT model.

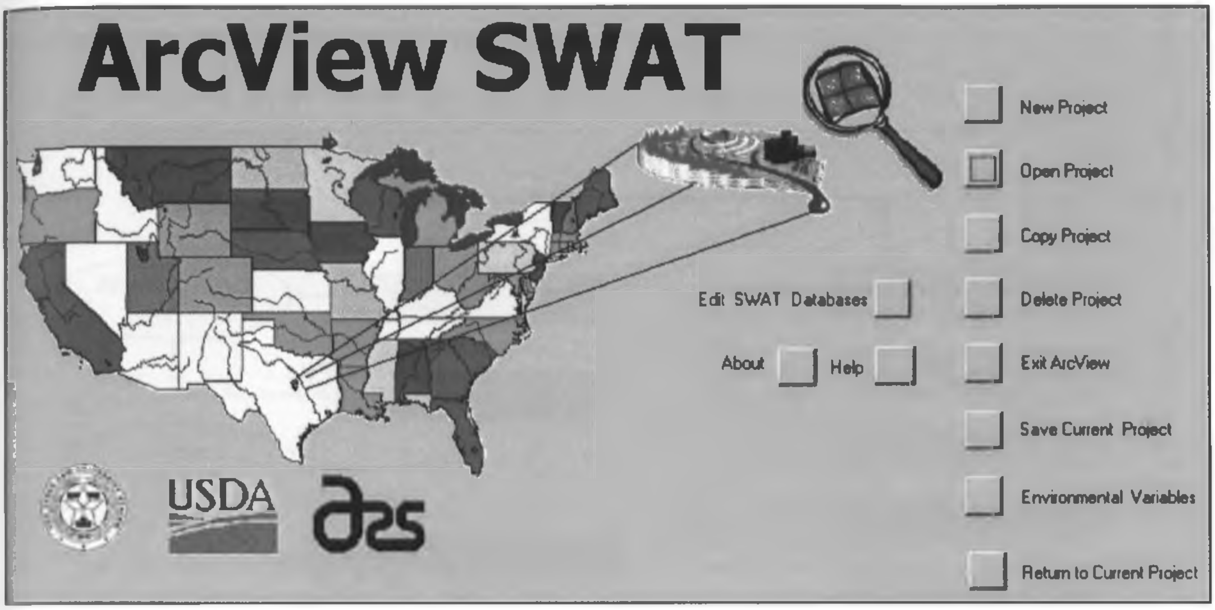


Figure 2.3: Map showing the main interface screen once AVSWAT is loaded in ArcView.

2.5.3 SWAT-CUP

SWAT-CUP (SWAT Calibration and Uncertainty Procedures) is designed to integrate various calibration/uncertainty analysis programs for SWAT using the same interface. Currently the program can run SUFI2 (Abbaspour et al., 2007), GLUE (Beven and Binley, 1992), and ParaSol (van Griensven and Meixner, 2006). To create a project, the program guides the user through the input files necessary for running a calibration program. Each SWAT-CUP project contains one calibration method and allows user to run the procedure many times until convergence is reached. User can save calibration iterations in the iteration history for later use. Also we have made it possible to create graphs of observed and simulated data and the predicted uncertainty about them.

The program SWAT-CUP coupling various programs to SWAT has the general concept shown in Fig.2.4 below. The steps are: 1) calibration program writes model parameters in model.in, 2) swat_edit.exe edits the SWAT's input files inserting the new parameter values, 3) the SWAT simulator are run, and 4) swat_extract.exe program extracts the desired variables from SWAT's output files and write them to model.out. The procedure continues as required by the calibration program.

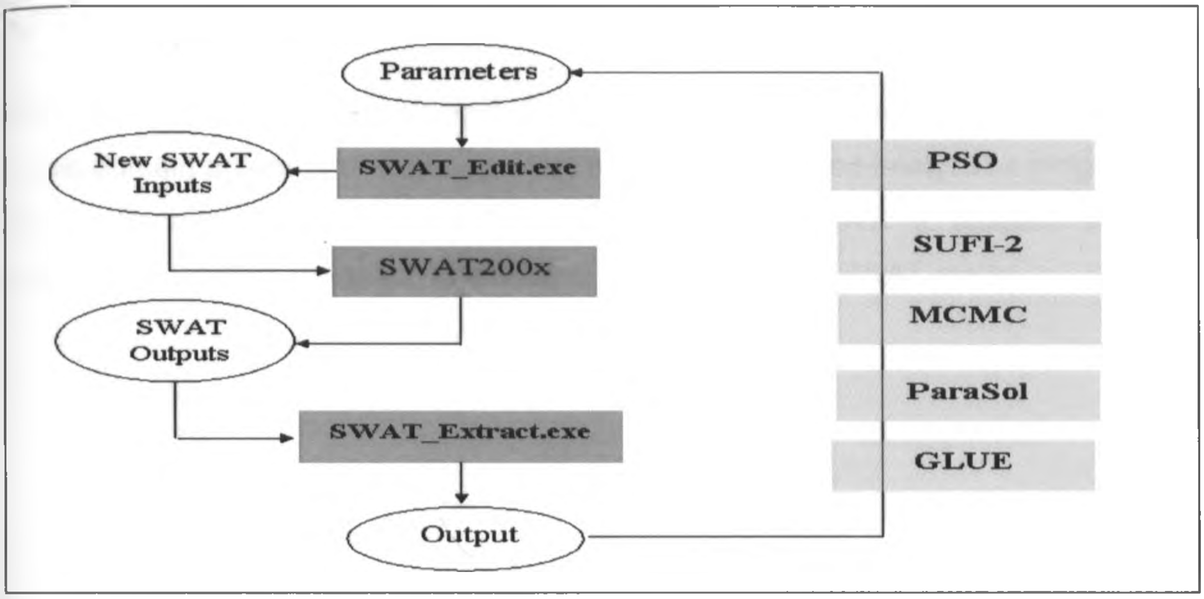


Figure 2.4: Program structure for SWAT-CUP

2.6 Summary on literature review

There are still great opportunities in the use of GIS in environmental engineering aspects. This is in respect to the statistical routines which have been developed but which are still lacking in their accuracies especially during sampling processes. Development of a Latin hypercube sampling technique within a GIS environment can solve this problem.

LHS is a good sampling protocol. It is a robust and efficient statistical framework that has been widely used to guide the determination of sample sizes given a set of constraints especially, when combined with GIS software suçh as ArcView. It can improve sample size selection and subsequent unbiased strategic location of the samples in the landscape given set of geo-referenced constraints.

ArcView is versatile GIS software which can be used for random selection of points/samples in a feature space. It allows for programs to be written as add-on codes. It also has the potential for hosting LHS in a GIS environment.

SWAT-CUP is a computer program for calibration of SWAT models. SWAT-CUP is a public domain program, and as such may be used and copied freely. The program links GLUE, ParaSol, SUFI2, and MCMC procedures to SWAT. It enables sensitivity analysis, calibration, validation and uncertainty analysis of a SWAT model.

3: MATERIALS AND METHODS

3.1 Latin hypercube sampling (LHS) theory

In order to understand how LHS can aid unbiased sampling in a geographic space, its theory was first explored. This theory was explored beginning from the principle of statistical sampling. Basically, sampling principle involves choosing a certain number of samples from a given distribution so that all characteristics of the distribution are represented in the sample. Suppose a given study variable has a normal probability distribution such as shown in Fig. 3.1, then its sampling involves choosing *n* members of the distribution so that the *mean* of these samples is related to the mean of the entire population as follows;

$$\overline{X} = \mu + Z_{(\alpha)} * \sigma / \sqrt{n} \tag{3.1}$$

where μ and σ are the mean and standard deviation of the population and Z is a standard normal variate at 100(1-α)% confidence interval.

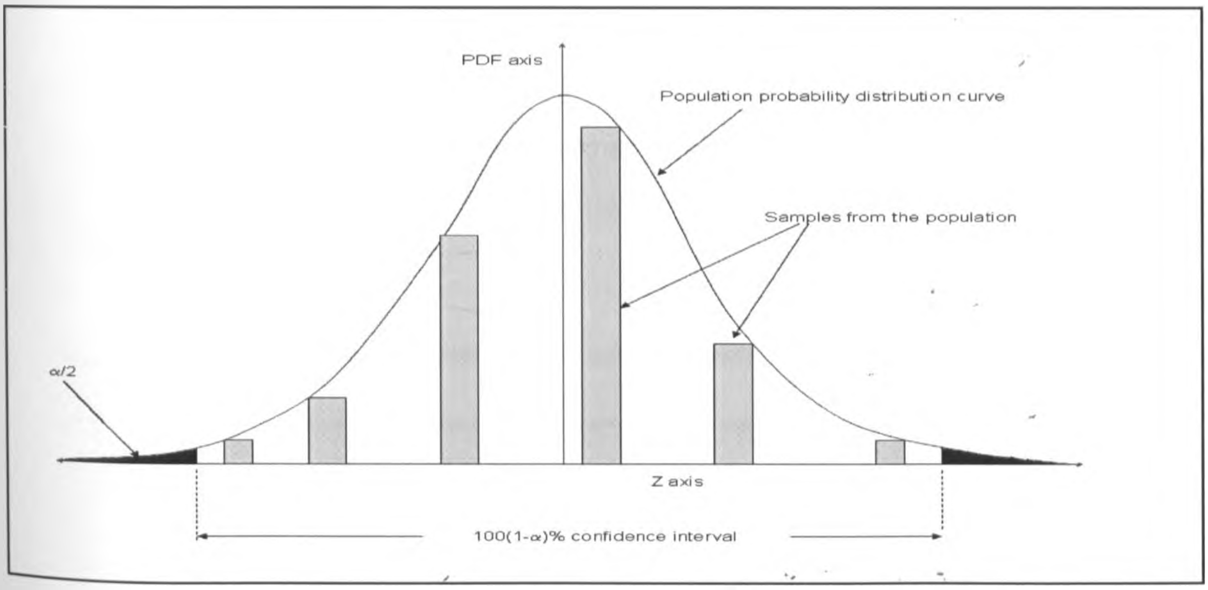

PDF axis
Population probability distribution curve
Samples from the population
α/2
Z axis
100(1-α)% confidence interval

Figure 3.1: Sampling from a normal probability distribution.

By using an acceptable sampling error, the number of samples is determined from the modification of equation (3.1) as shown in equation (3.2).

$$n = \left(\frac{Z_{(\alpha)} * \sigma}{Error} \right)^2 \tag{3.2}$$

where sampling $Error = (\overline{X} - \mu)$.

The number of samples obtained from equation 3.2 is the theoretical sample size which keeps the sampling error below $(\overline{X} - \mu)$ with 100(1-α) % confidence interval (Kottegoda and Rosso, 1998).

In practice, the number of samples is usually dictated by factors such as sampling cost, study area, sampling time, etc. It is therefore not uncommon to find many sampling studies with a priori sampling size (*n* in this case). Hence, the problem of sampling shifts to how to determine accurate positioning (in the overall population probability distribution) of the chosen samples so that they make an unbiased representation of the entire landscape.

In LHS sampling, this problem is circumvented by assuming that the population comprises of *k* as discrete univariate normal distributions (intersecting orthogonally to each others) so that they form a *k*-hypercube. As an example, if *k*=2, the hypercube appears as shown in Fig.3.2.

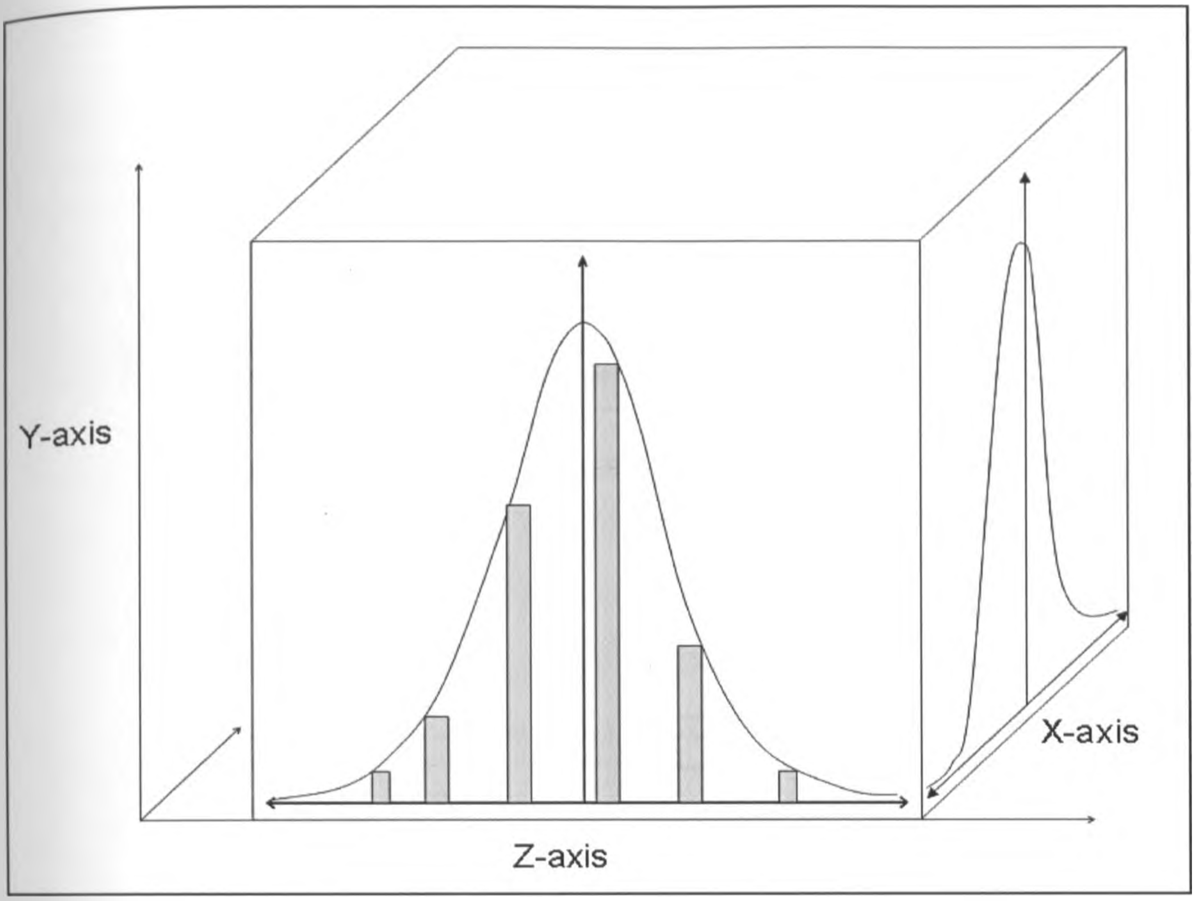


Figure 3.2: Example of hypercube for sampling from two populations.

Using Fig. 3.2 as an illustration, LHS seeks to choose *n* samples from the two distributions (e.g. elevation and land use) such that for each distribution there are *n* equally probably strata to represent the distributions. Statistically, this can be looked at as dividing the distribution (or cumulative distribution) of each population into *n* equal intervals as shown in Fig. 3.3. The *n* strata from each distribution are then bound together based on some rules (McKay et al., 1979).

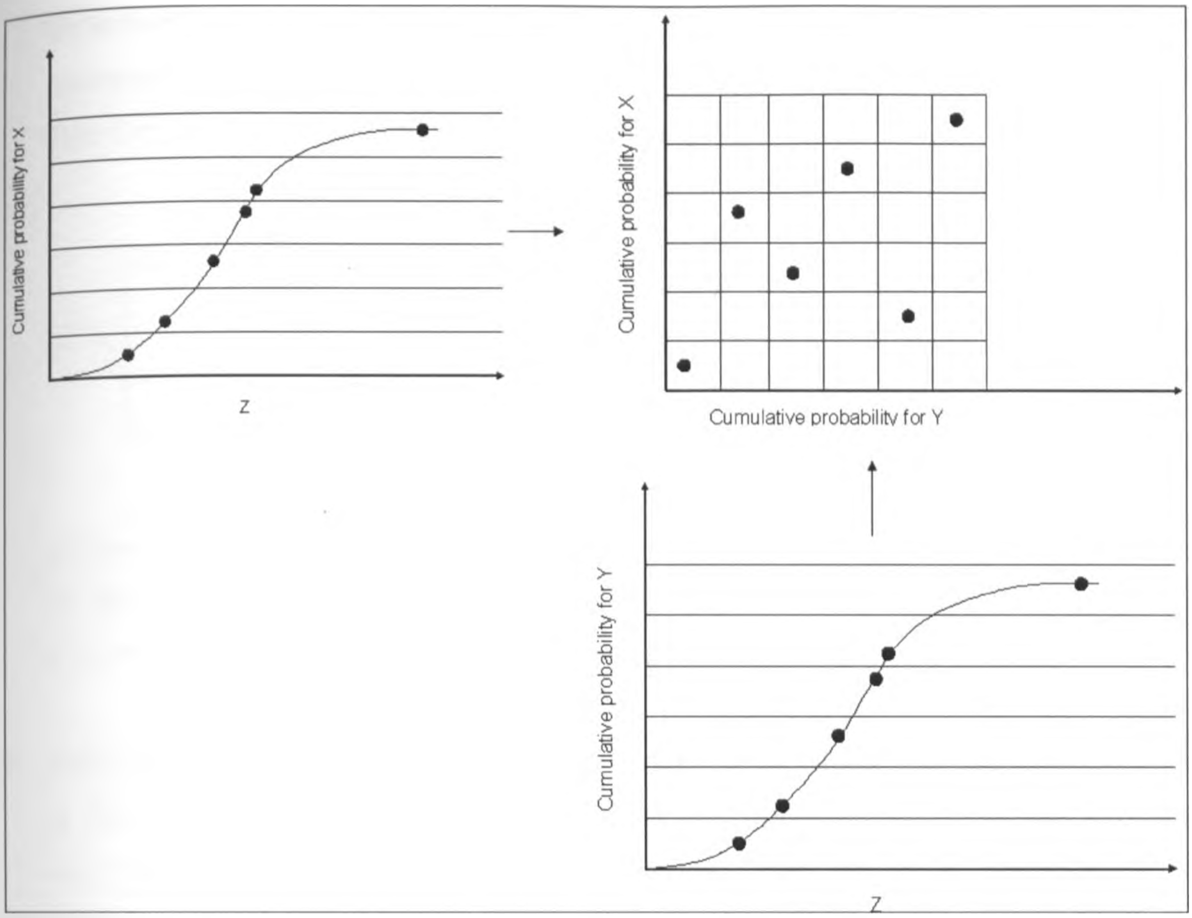

Cumulative probability for X
Z
Cumulative probability for X
Cumulative probability for Y
Cumulative probability for Y
Z

Figure 3.3: Example for five LHS samples from two populations.

In order to implement the above LHS for more than two variables, the following algorithm was designed in this study:

1. Determination of the sampling constraints. For example, if sampling interest is to have 30 locations in a study area and targets lowlands (e.g. elevation < 1000 m), grasslands, and 500 m from the main road; then the LHS constraints would include: sample size, *n* = 20, lowlands (from a DEM), grasslands (from a land use map), and a buffer region of 500 m from the main road. These constraints (or sampling variables) definitely have probability distributions which can be generated and sampled as shown in Fig. 3.3.

Determination of the distribution of the above constraints (or variables): Where necessary, the distributions are converted to multivariate normal distribution using Box-Cox transformations as given in equation (3.3).

$$y^T = \begin{cases} \dfrac{y^l - 1}{l} & l \neq 0 \\ \\ log_e(y) & l = 0 \end{cases} \tag{3.3}$$

where;

l is the Box-Cox index of transformation (Kottegoda and Rosso, 1998).

y^T is the transformed y variable.

2. Estimation of the probability distribution: Since the variables may have perturbations in their probability functions, it is better that the probability functions are simulated and the results used to select equal strata. The simulated distributions are statistically tractable and can expedite the choosing of *n* strata better than the actual distributions. In this study, the simulations for the distributions were done using Monte Carlo simulations for an empirical normal distribution function as shown in equation (3.4). The normal distribution function was used since in equation (3.3) they automatically assume normal distribution after the transformation.

$$f'(x) = \left(\frac{r_n}{n}\right) + \left(\frac{i - 0.5}{n}\right) \tag{3.4}$$

where;

r_n are random numbers conforming to the normal distribution.

i is the rank value of transformation.

3. Transformation of the above probability function into a distribution function: this was done by integrating the above empirical normal multivariate probability functions. As an illustration; suppose that the normal multivariate probability function is given by equation (3.5);

$$f(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^p \, |\Sigma|}} \exp[-0.5(\mathbf{x} - \boldsymbol{\mu})^T \Sigma(\mathbf{x} - \mu)] \tag{3.5}$$

where Σ is a (*p* x *p*) variance-covariance symmetric positive definite matrix, **x** is the vector of LHS variables, and **μ** is the vector of mean of the LHS variables. Then, the integral of the distribution function is given using equation (3.6).

$$F(\mathbf{x}) = \int \frac{1}{\sqrt{(2\pi)^p \, |\Sigma|}} \exp[-0.5(\mathbf{x} - \boldsymbol{\mu})^T \Sigma(\mathbf{x} - \mu)] d\mathbf{x} \tag{3.6}$$

The output of the above integral (also known as cumulative distribution function) is then divided into *N* equal strata from which *n* are chosen. This then becomes a case of optimization problem: where given *N* strata, *n* are chosen randomly while retaining probability distribution of *N* strata. In this study, this was hypothesized as follows:

a. Rank the simulated probabilities in ascending order

b. Choose the first rank

c. Choose $\left(\frac{N}{n} - 1\right)^{th}$ sample from the first choice in (*b)* above

d. Repeat step (*c*) above consecutively till the rank-size is exhausted

The output from above loop produces *n* from *N* strata of each variable (or LHS constraint).

4. Merging strata allocations: The *n* values obtained for each variable are then paired randomly throughout their cumulative probability range (as shown in Fig. 3.3).

3.2 Implementing LHS theory in ArcView

LHS theory described in section 3.1 was written in avenue programming language for implementation in ArcView. Avenue is ArcView's object-oriented programming (OOP) or scripting language for ArcView. Its programming techniques include features such as information hiding, data abstraction, encapsulation, modularity, polymorphism, and inheritance. In ArcView, objects include "View", "Table", etc. OOP in ArcView involves sending requests to objects and the two separated by period. For example, a request to display a view as written in OOP language appears as follows:

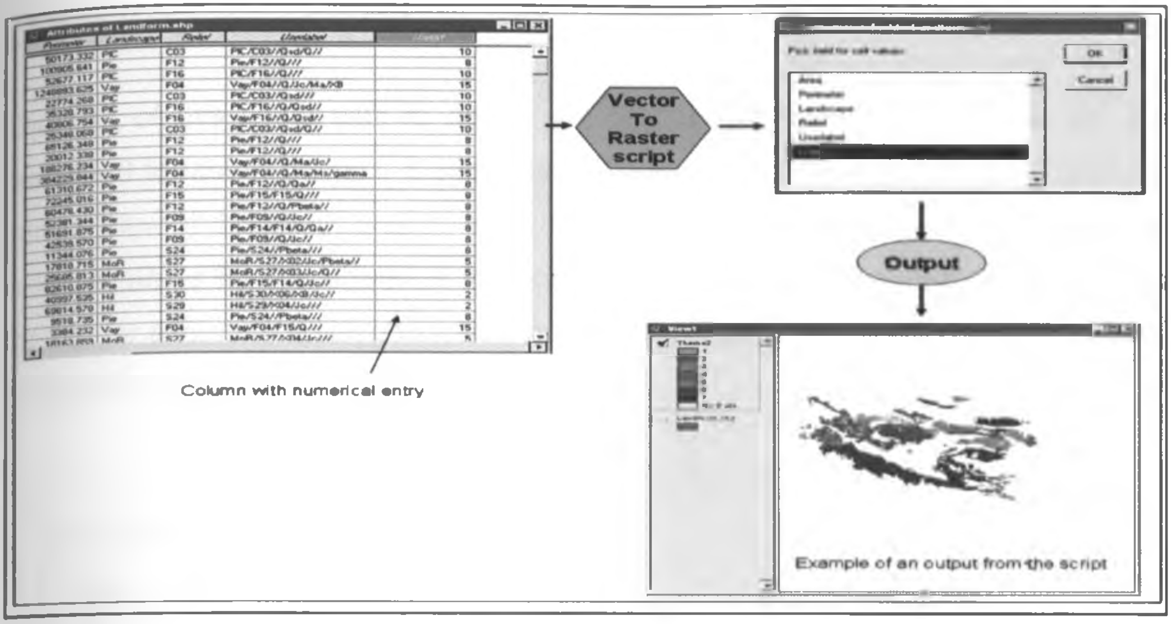
theView.GetDisplay

where the object (***theView***) precedes the request (***GetDisplay***). Hutchinson and Larry (1999) have described OOP in detail.

In this study, ArcView scripts written in OOP were used to implement the LHS steps as described in the following sub-sections.

3.2.1 Script to convert vector to raster

This script was written for LHS cases where users have some LHS constraints as vector GIS layers, for example, in choosing certain polygons in a land use map or using a buffer zone from a given spot/line/road. The input for this script is a polygon vector (e.g. land use map, buffer zone, etc) in which the *Attribute table* has a field/column with numerical entry to be used as the pixel output value in the final raster map (Fig.3.4). The features of the script include the provision for the user to specify the output cell size and coverage (as shown in Fig. 3.5).



Attributes of Landform.shp

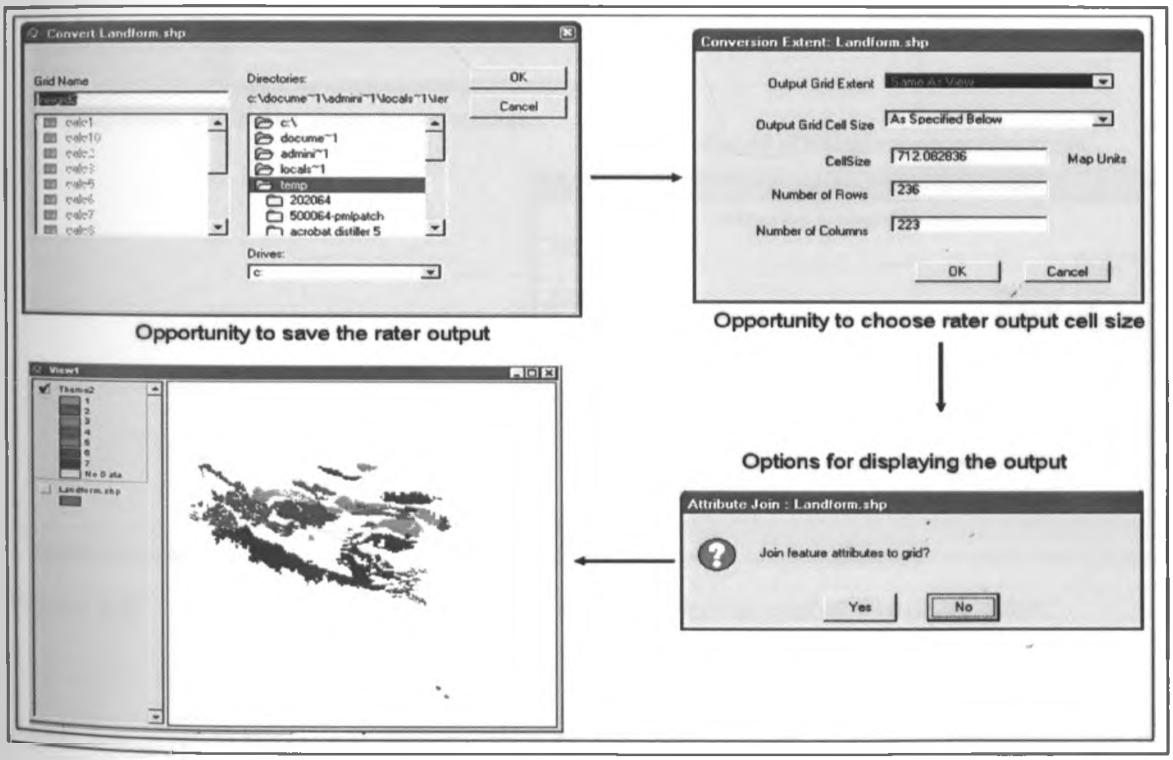
Perimeter	Landscape	Relief	Unitlabel	[illegible]
50173.332	PIC	C03	PIC/C03//Qsd/Q//	10
100905.641	Pie	F12	Pie/F12//Q///	8
52677.117	PIC	F16	PIC/F16//Q///	10
1248893.625	Vay	F04	Vay/F04//Q/Jc/Ma/XB	15
22774.268	PIC	C03	PIC/C03//Qsd///	10
35320.793	PIC	F16	PIC/F16//Q/Qsd//	10
40906.754	Vay	F16	Vay/F16//Q/Qsd//	15
25349.068	PIC	C03	PIC/C03//Qsd/Q//	10
65126.348	Pie	F12	Pie/F12//Q///	8
20012.339	Pie	F12	Pie/F12//Q///	8
188276.234	Vay	F04	Vay/F04//Q/Ma/Jc/	15
364225.644	Vay	F04	Vay/F04//Q/Ma/Hs/gamma	15
61310.672	Pie	F12	Pie/F12//Q/Qa//	8
72245.016	Pie	F15	Pie/F15/F15/Q///	8
60478.430	Pie	F12	Pie/F12//Q/Pbeta//	8
52381.344	Pie	F09	Pie/F09//Q/Jc//	8
51691.875	Pie	F14	Pie/F14/F14/Q/Qa//	8
42538.570	Pie	F09	Pie/F09//Q/Jc//	8
11344.076	Pie	S24	Pie/S24//Pbeta///	8
17810.715	MoR	S27	MoR/S27/X02/Jc/Pbeta//	5
29595.813	MoR	S27	MoR/S27/X03/Jc/Q//	5
82610.875	Pie	F15	Pie/F15/F14/Q/Jc//	8
40597.525	Hil	S30	Hil/S30/X06/XB/Jc//	2
69614.570	Hil	S29	Hil/S29/X04/Jc///	2
9518.735	Pie	S24	Pie/S24//Pbeta///	8
3384.232	Vay	F04	Vay/F04/F15/Q///	15
18163.869	MoR	S27	MoR/S27/X04/Jc///	5

Pick field for cell values:

- Area
- Perimeter
- Landscape
- Relief
- Unitlabel

OK
Cancel

Figure 3.4: Example of vector-to-raster scripting steps and output.



Convert Landform.shp

Grid Name: [illegible]

- calc1
- calc10
- calc2
- calc3
- calc5
- calc6
- calc7
- calc8

Directories:
c:\docume~1\admini~1\locals~1\ter

- c:\
- docume~1
- admini~1
- locals~1
- temp
- 202064
- 500064-pmlpatch
- acrobat distiller 5

Drives: c:

OK
Cancel

Opportunity to save the rater output

Conversion Extent: Landform.shp

Output Grid Extent: Same As View
Output Grid Cell Size: As Specified Below
CellSize: 712.082836 Map Units
Number of Rows: 236
Number of Columns: 223

OK
Cancel

Opportunity to choose rater output cell size

Options for displaying the output

Attribute Join : Landform.shp

Join feature attributes to grid?

Yes
No

Figure 3.5: Features of vector-to-raster script in the LHS code.

3.2.2 Script for generating histogram

The aim of this script was to depict the probability distribution of raster (or vector converted to raster) themes. It generates a histogram for the active raster theme in the current view and displays it as new bar chart. The color scheme used to create the chart is the same as for the legend of the active theme. The input for this script is an active theme, for example, a polygon with a numerical field in its database table (Fig. 3.6). An output example from the script is shown below.

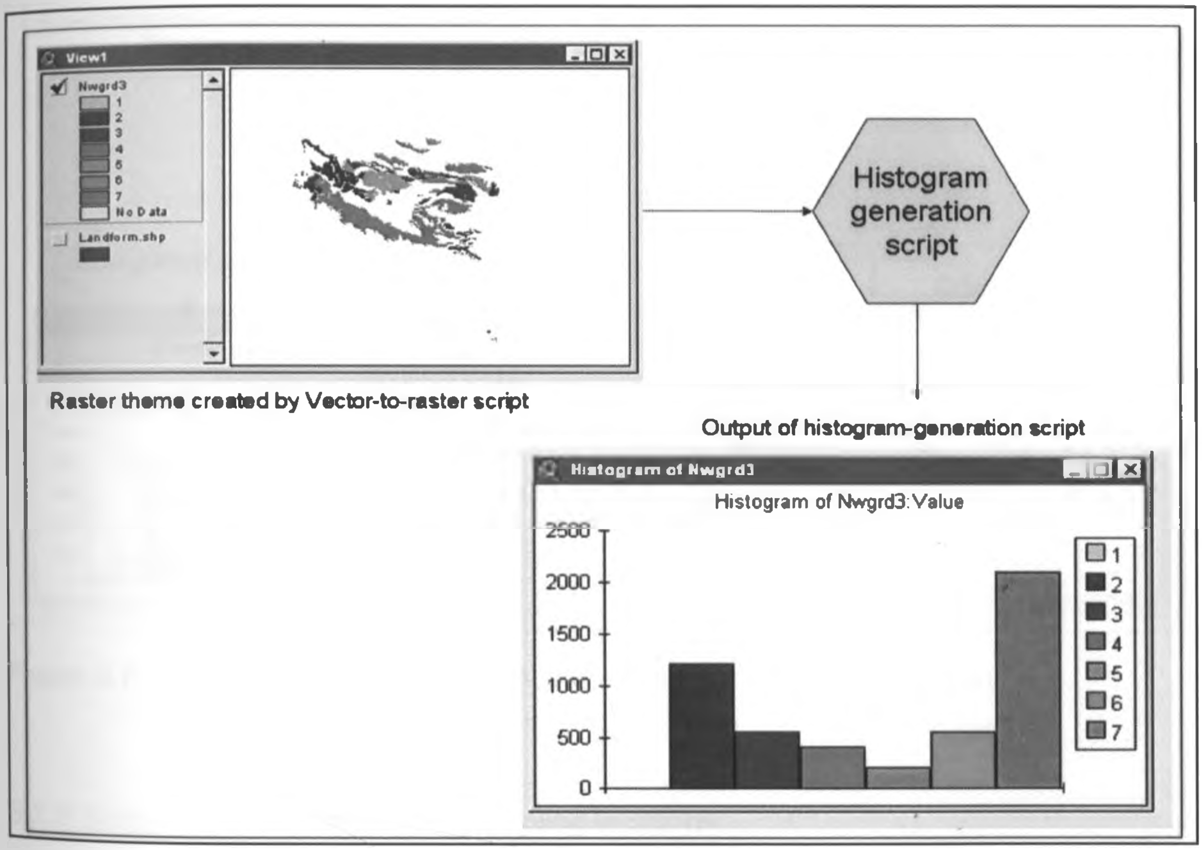
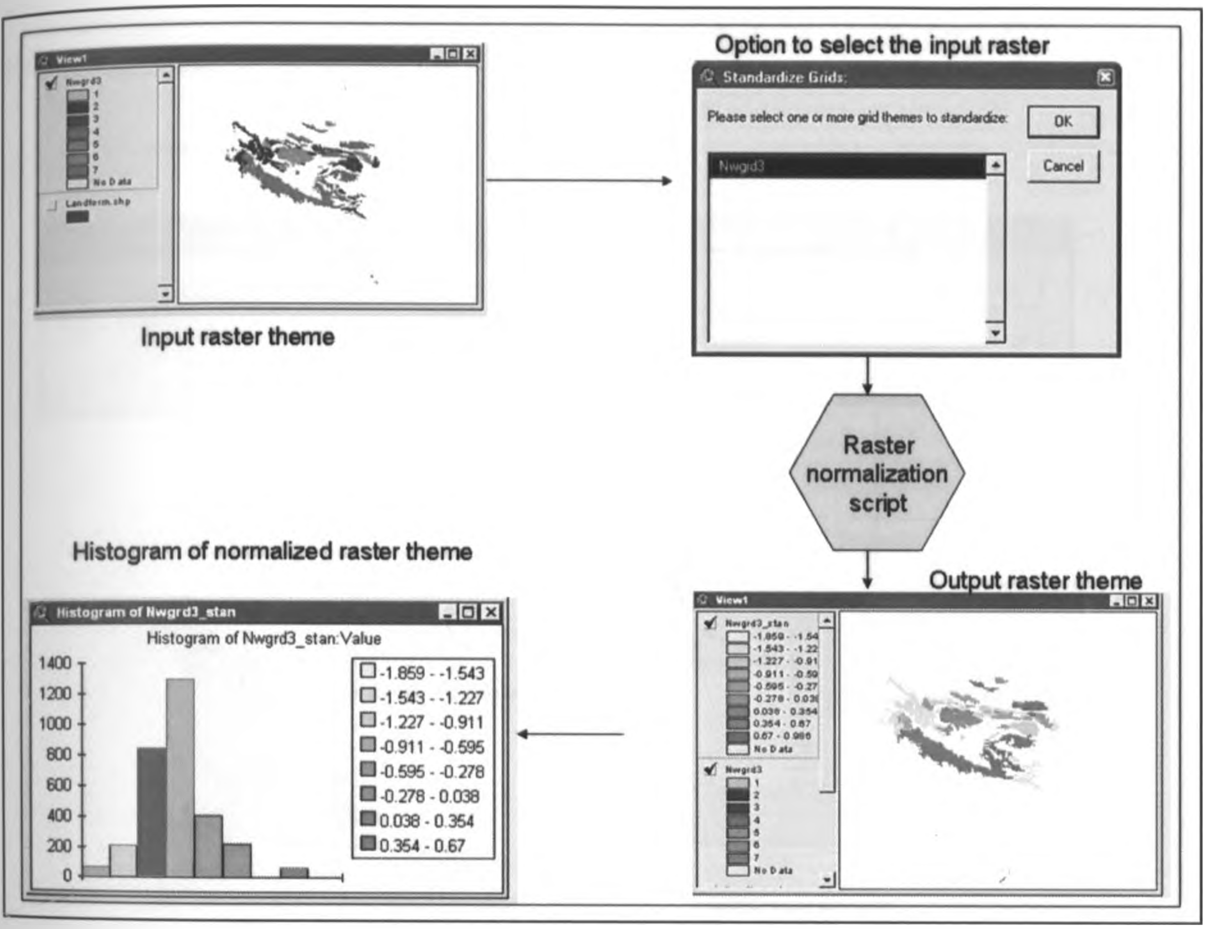

View1
Nwgrd3
1
2
3
4
5
6
7
No Data
Landform.shp
Histogram generation script
Raster theme created by Vector-to-raster script
Output of histogram-generation script
Histogram of Nwgrd3
Histogram of Nwgrd3:Value
2500
2000
1500
1000
500
0
1
2
3
4
5
6
7

Figure 3.6: Histogram generation output with non-normal probability distribution.

From the histogram generated by this script, sometimes it may suffice that the raster theme be transformed into a normal probability distribution. A script was also written for cases requiring such normalization step. For example, the raster image created in Fig.

3.5, which had non-normal distribution (Fig. 3.6), was transformed into a normal probability distribution as shown in Fig. 3.7.



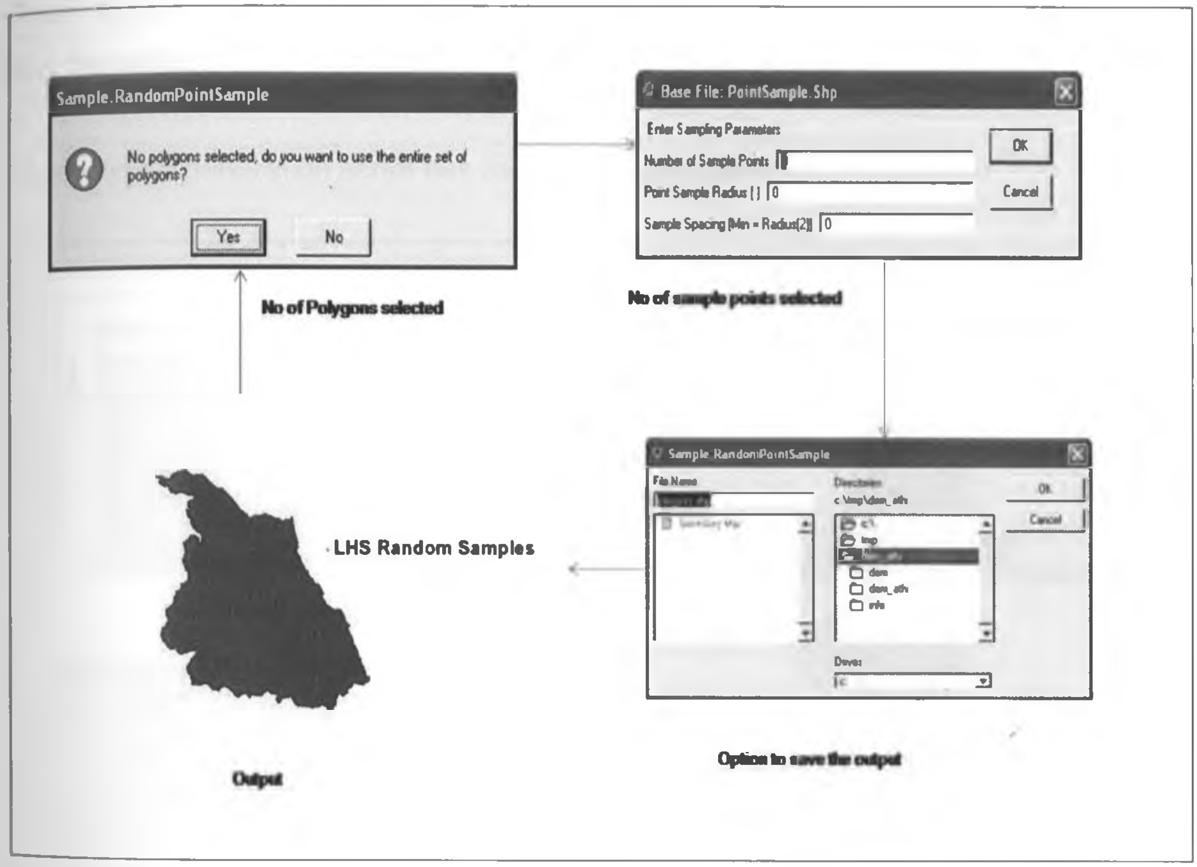
Please select one or more grid themes to standardize:

Figure 3.7: Histogram generation output with normal probability distribution.

3.2.3 Script for random search of sampling locations

This script was used to create a set of random points within a specified stratum of the probability distribution obtained from the above script outputs. The number of strata was left open to be specified by the user (as number of sampling points desired). In order to search through each strata of the probability, the histogram generated from the above step is converted back into polygons so that the search is contained within the polygon boundary. The use of polygons was preferred since they have hard boundaries compared to raster pixel boundaries. A script for converting the histogram classes into polygons was written for this purpose. Once established, the polygons are supplied as input for searching the optimum location for selected sample size.

The random-search script has the following features;



Sample.RandomPointSample

No polygons selected, do you want to use the entire set of polygons?

Base File: PointSample.Shp

Enter Sampling Parameters

Number of Sample Points

Point Sample Radius ( ) 0

Sample Spacing (Min = Radius(2)) 0

Sample.RandomPointSample

File Name

Directories: c:\tmp\dem_ath

Drives: c:

Figure 3.8: Random search sample features.

3.2.4 Script for converting histogram into polygons

A script for converting histograms into polygons was also written to help the researcher/field scientist to be able to carry out studies with raster data as the input. For example, a researcher/field scientist carrying out a study using DEM as the input data must generate histograms first. These histograms are then split into bins. Reclassification of the raster input data using the classes from the histogram is done and thereafter they are combined. The output from this script is as shown in Fig. 3.9.

The polygon generation script has the following features;

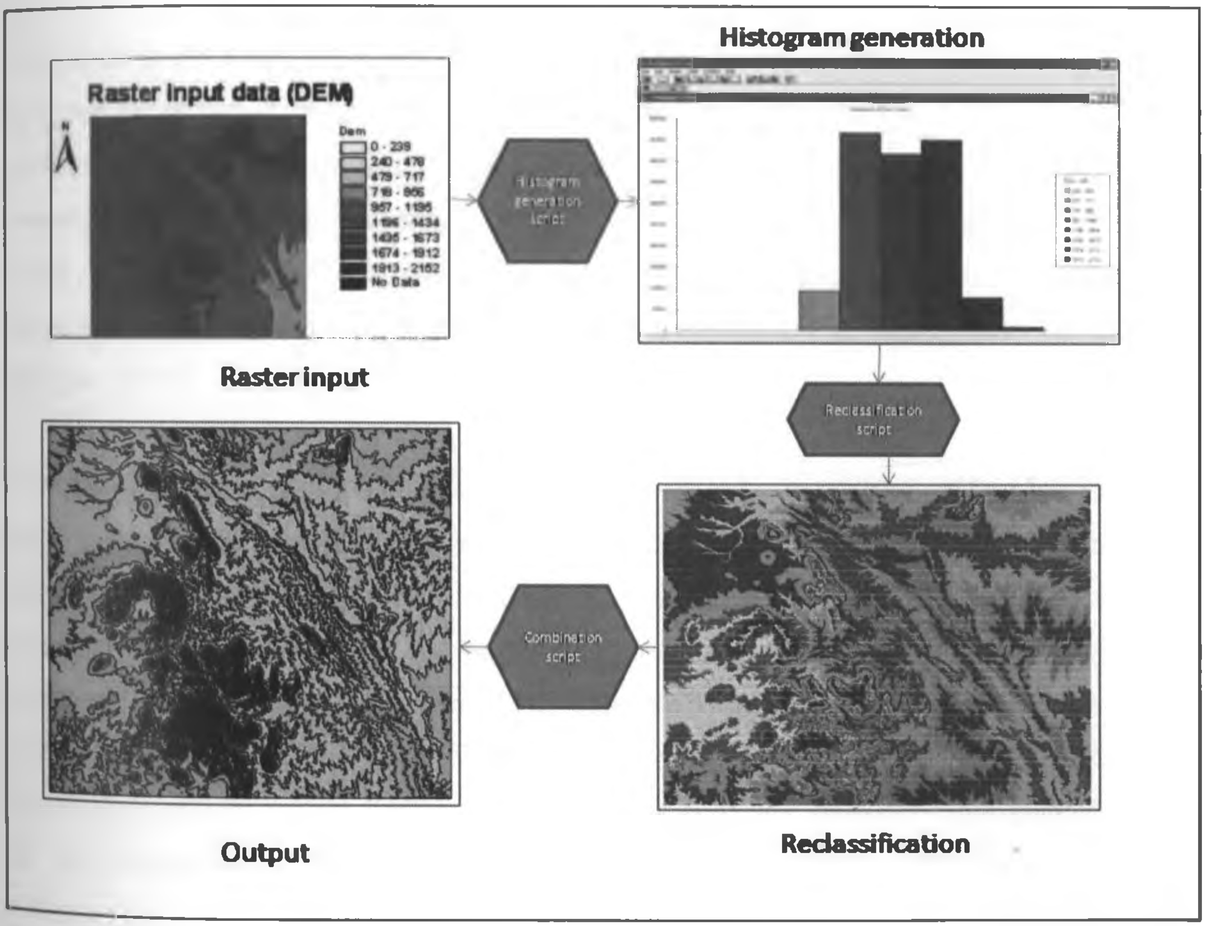


Figure 3.9: Polygon generation output from raster data.

The above scripts were then integrated into one unit using ***extbuild.apr script*** available in ArcView Developers' kit (www.esri.com) to produce the LHS extension. This script was capable of building the skeletons for a program to run in ArcView GIS software. The built skeleton was customized to provide for input dialogue, GUI applications, output formatting and opportunities for program update. The final product was then saved as an ArcView extension named ***LHS.avx***. It can be uploaded in any ArcView program and used just like other ArcView extension.

3.3 Testing the LHS extension

The LHS code was tested for its ability to exhaustively sample a feature space and geographic space. In testing the code for its adequacy in sampling a feature space, the criterion was to compare the population distribution of the landscape attribute and the corresponding distribution of the attribute at the generated sample locations. In order to achieve this, two histograms (one for the population and another for the samples) were compared in what is known as back-to-back comparison of histograms (Omuto and Vargas, 2009).

For example, suppose that 300 sampling points have been generated by LHS extension for a given study area and the interest is to compare the landscape elevation of the study area as obtained from the DEM to the landscape elevation at the sampled locations, then back-to-back comparison of these two quantities as shown in Fig. 3.11 gives the test for adequacy of the LHS extension representing elevation as a feature space. If the distributions of these two quantities were the same, then the bins of the histograms should reflect themselves along the zero line dividing them (Fig. 3.11). As for the example in Fig. 3.11 some bins in the population distribution were not represented in the sampling distribution. However, in general, the two distributions statistically compare, which implies that the sampling was very close to representing the population characteristics.

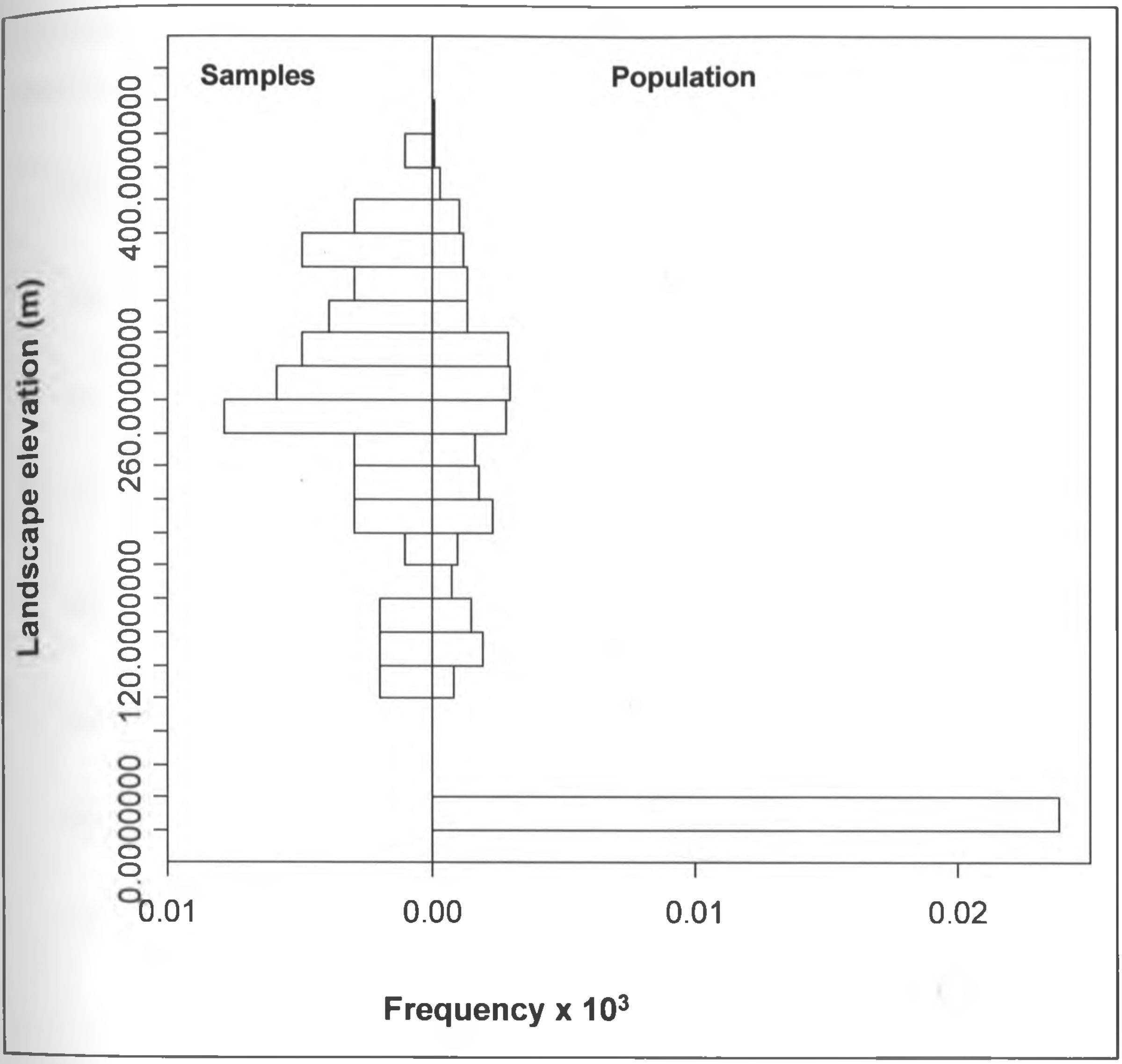


Figure 3.10: Random search sample features.

The back-to-back principle shown in the Fig. 3.10 was also used to compare sampling by LHS code and other popularly used protocols (such as random sampling and stratified random sampling). The testing was carried out for the Upper Athi River Basin in Eastern Kenya. The study area is shown in Fig. 3.13 (Omuto, 2008). Two feature spaces were used: elevation and remotely sensed Normalized Difference Vegetation Index (NDVI). In each sampling protocol tested, 20 points were generated and the sampling protocol reflects the population distribution taken as the best alternative. The population distributions of these two quantities (i.e. NDVI and elevation) in the study area are given in Figs. 3.11 and 3.12.

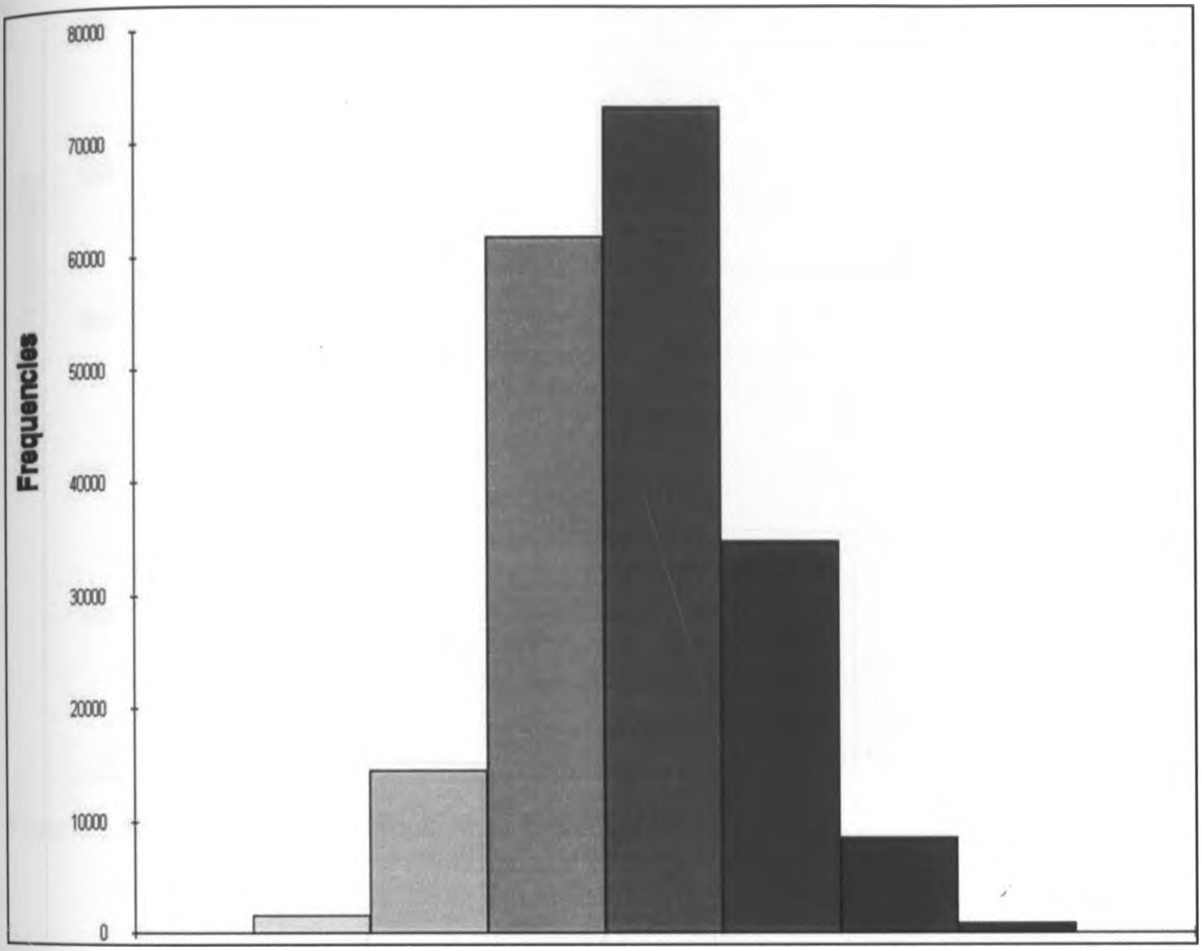


Figure 3.11: Probability distributions of NDVI.

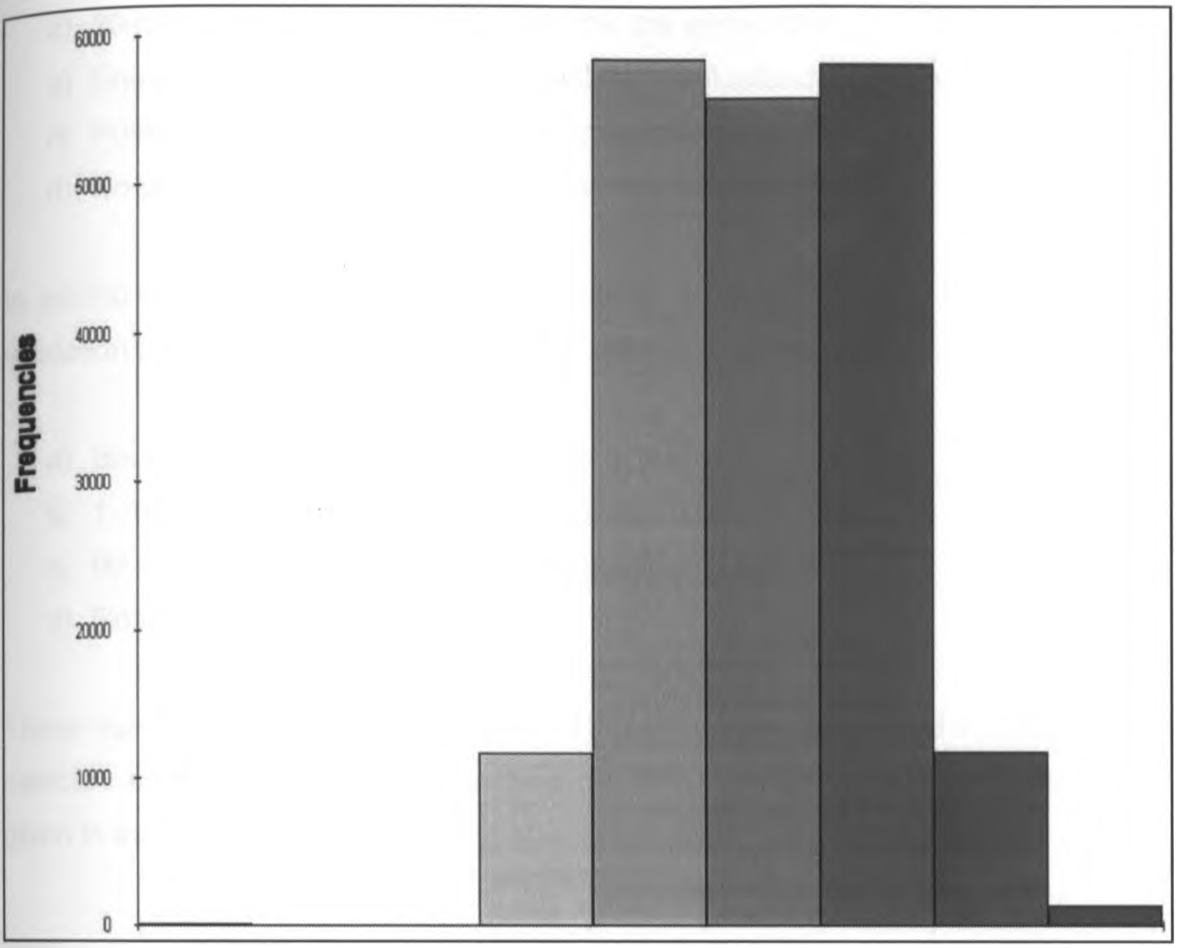


Figure 3.12: Probability distributions of elevation.

3.4 Application of LHS extension in environmental engineering

Application of the LHS extension in environmental engineering was tested in a study of loss of topsoil in the Upper Athi River Basin, Kenya (Omuto, 2008). In this study, the loss of topsoil for the whole study area was obtained and was validated using 20 randomly selected points in the study area. The validation was done according to shrub-mound method (Omuto and Vargas, 2009) at about 20 m from the GPS coordinates of the randomly selected validation points. The selection of the validation points was achieved using the LHS extension. The following constraints guided the choosing of the validation points;

a) 20 random points to be selected within the study area
b) Equal representation of upland, midland and lowland slope zones
c) Points should be at least 500 m from the main road
d) No point should fall in urban centres and water bodies

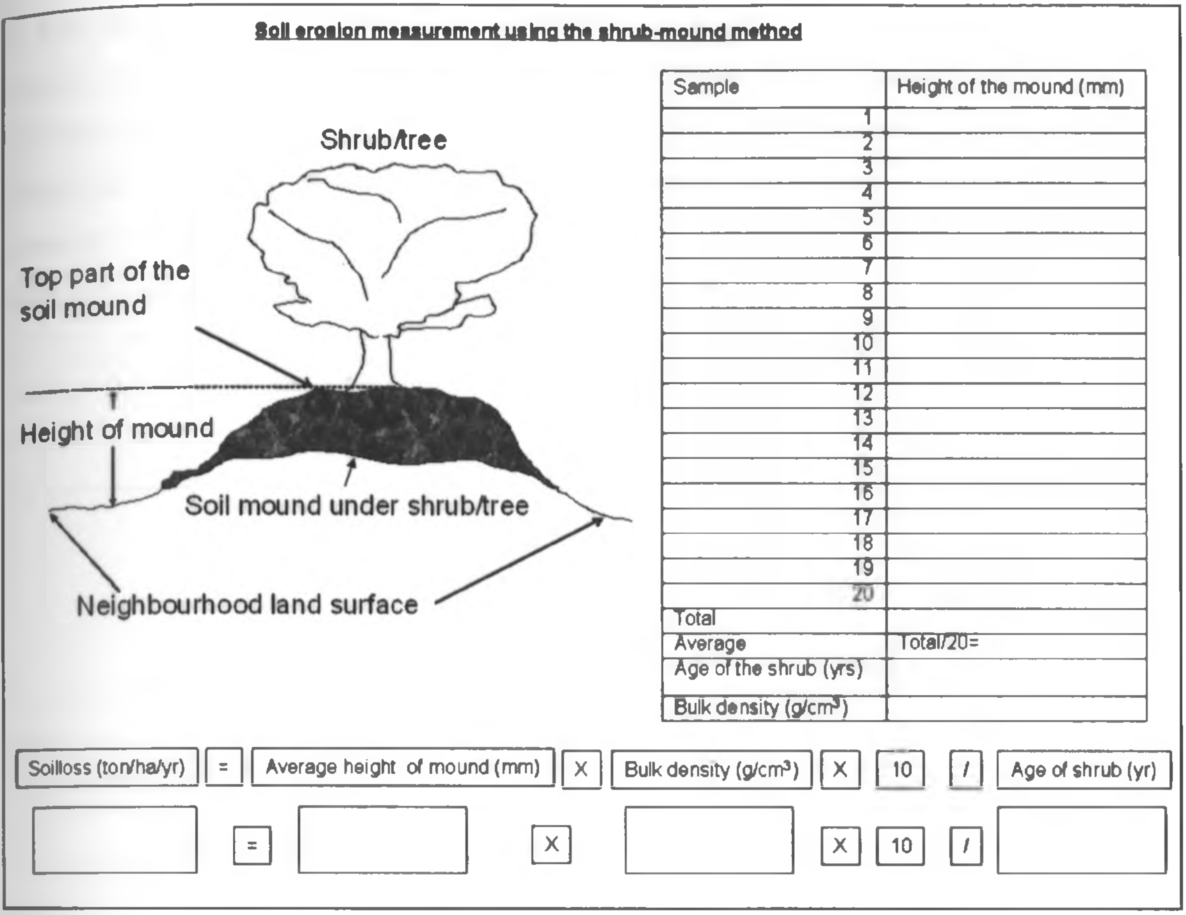
In addition, the following GIS inputs were used to implement the LHS selection of the validation points;

a) Boundary of the study area (see Fig. 3.14)
b) 1: 100,000 Land cover map from Omuto (2008)
c) 90-m spatial resolution DEM (from http://srtm.usgs.gov)
d) Road network for the study area

These inputs were processed using the LHS extension as described in section 3.2. The selected points were visited in the field on 15th June 2009 and topsoil measured as given in equation 3.7.

$$soilloss = \frac{height * 13}{age} \tag{3.7}$$

where *soilloss* is the rate of topsoil loss in tonnes/ha/yr, *height* is the height of the mound underneath the shrub (Fig. 3.13) and *age* is the approximate age of the shrub (Omuto and Vargas, 2009). The shrub-mound method for measuring topsoil loss assumes the soil under a shrub to benefit from the protective cover of the shrub against agents of erosion over time; thus, forming a mound with respect to soil level in the neighbourhood. The height difference between the top part of the mound and soil level in the neighbourhood represents the amount of soil lost during the age of the shrub (Stocking and Murnaghan, 2001; Omuto and Vargas, 2009).


Shrub/tree
Top part of the soil mound
Height of mound
Soil mound under shrub/tree
Neighbourhood land surface

Soil erosion measurement using the shrub-mound method

Sample	Height of the mound (mm)
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
Total	
Average	Total/20=
Age of the shrub (yrs)	
Bulk density (g/cm³)	

Soilloss (ton/ha/yr) = Average height of mound (mm) X Bulk density (g/cm³) X 10 / Age of shrub (yr)

□ = □ X □ X 10 / □

Figure 3.13: Soil erosion measurement using the shrub-mound method.

The topsoil loss obtained from the validation points were then compared to the topsoil loss given by Omuto (2008). The comparison was done using the Nash-Sutcliffe coefficient of efficiency (Nash and Sutcliffe, 1970) given by;

$$R^2 = 1 - \frac{\sum_{i=1}^{20}(L_{vi} - L_{0i})^2}{\sum_{i=1}^{20}(L_o - \overline{L_o})^2} \qquad (3.8)$$

Where R^2 is the coefficient of efficiency, L is the rate of topsoil loss (tonnes/ha/yr), v is the validated and o is the observed [(from Omuto (2008)] rate of topsoil loss. According to the Nash-Sutcliffe coefficient of efficiency, a value of 1 ($R^2=1$) corresponds to a perfect match between prediction and observed data. Efficiency of 0 ($R^2=0$) indicates that the predictions are as accurate as the mean of the observed data, whereas an efficiency less than zero ($-\infty<R^2<0$) occurs when the observed mean is a better predictor than the model (Nash and Sutcliffe, 1970). Essentially, the closer the prediction efficiency is to 1, the more accurate the model is.

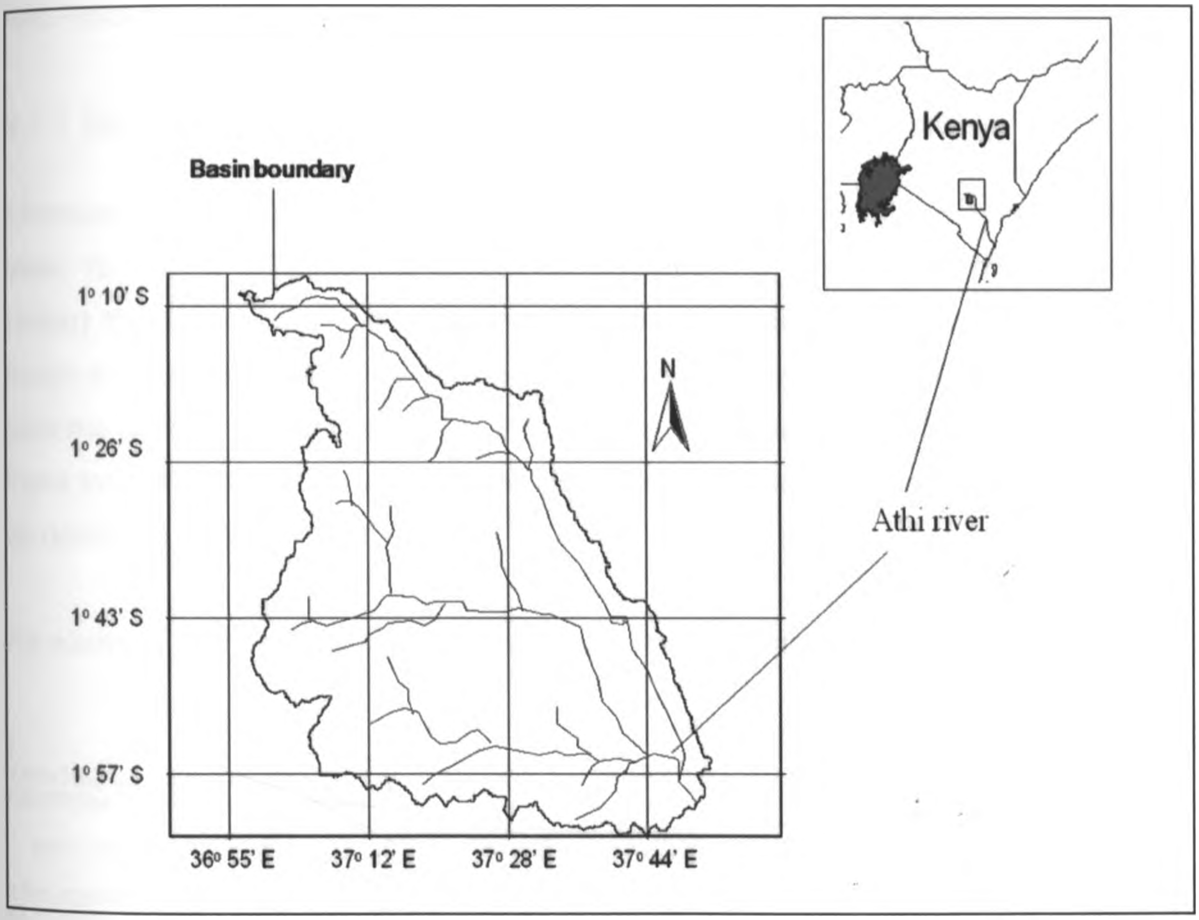


Figure 3.14: The Upper Athi River Basin in Eastern Kenya.

4: RESULTS AND DISCUSSION

4.1 ArcView scripts for implementing LHS

Four avenue scripts were developed to facilitate implementation of LHS in ArcView GIS: a script for generating histograms, a script for converting the histogram bins into polygon shapefiles, a script for optimizing sampling locations within the polygons given a set number of sampling points and a script for converting vector to raster (for cases with input constraints in vector data formats).

4.1.1 Script for generating histogram

Histogram generation script develops a histogram for the active theme in the active view. Its' aim is to depict the probability distribution of raster (or vector converted to raster) themes. A new chart document is created to display the histogram. With this script a temporary file is created to store classes to create the histogram. The color scheme used to create the chart is the same as for the legend of the active theme. The input for this script is an active theme, for example, a polygon with a numerical field in its database table.

An example is shown below of a sample script for generating histograms.

```
{
theView=av.GetActiveDoc
theTheme=theView.GetActiveThemes.Get(0)

' Get the compthe usernts of the Legend that will be used to create the chart...

theLegend=theTheme.GetLegend
theSymbols=theLegend.GetSymbols
theClasses=theLegend.GetClassifications
theFieldName=theLegend.GetFieldNames.Get(0)

theVTab=theTheme.GetFTab
theField = theVTab.FindField(theFieldName)

' Create the new data file for the histogram...

outFName = av.GetProject.MakeFileName( theTheme.GetName, "dbf")
outFName = FileDialog.Put( outFName, "*.dbf", "Output Histogram File" )
if (outFName = Nil) then
```

```
  exit
end

newVTab=VTab.MakeNew( outFName, dBASE )
labelf=Field.Make( "Label", #FIELD_CHAR, 20, 0 )
countf=Field.Make( "Count", #FIELD_DECIMAL, 10, 0)
newVTab.AddFields( {labelf, countf} )


' Loop through the classes recording the ranges.
'
countlist = {}
for each c in theClasses
  countlist.Add(0)
end
numClasses = theClasses.Count


' Loop through the records recording which class they fall in.
'
for each rec in theVTab
  v = theVTab.ReturnValue(theField,rec)
  for each i in 0..(numClasses - 1)
    if (theClasses.Get(i).Contains(v)) then
      countlist.Set(i,countlist.Get(i)+1)
      break
    end
  end
end

}
```

When the above script is executed in ArcView, it generates a histogram with a probability distribution for the active layer. An output example from the above script is given in Fig. 4.1. This histogram was for a DEM for the Upper Athi River basin.

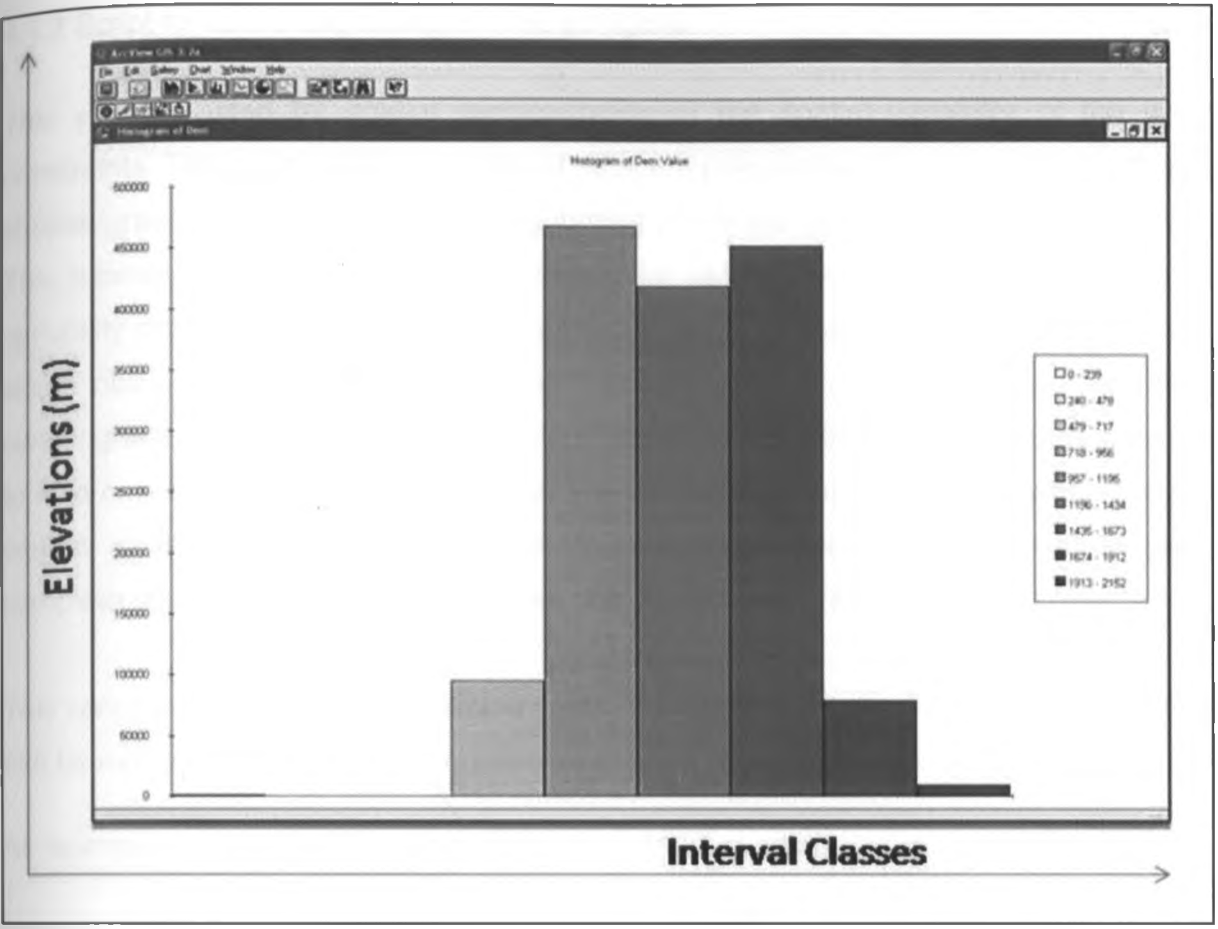


Figure 4.1: Output of histogram generation script.

The importance of histogram generation script is to support the development of probability distribution raster themes. From the above output (Fig. 4.1), it can be deduced that most parts of the study area fell within the elevations/heights of between 1195 m and 2152 m above the sea level. These areas can be described as the highlands within the study area. Some parts of this study area fell within elevations/heights of between 0 m to 957 m above the sea level. These areas can thus be described as lowlands.

The entire script is given in Appendix A.

4.1.2 Script for converting histogram into polygons

This script is used for spatial representation of the spatial variability of the input constraints. This script generates a set of adjacent polygons within the selected polygon shapes/graphics. In this script, the neighboring pixels are assigned the same polygons. This is because neighboring pixels exhibit similar characteristics within the same spatial variability range. A topology cleaning process is then carried out to remove minute and single pixels where they are collapsed within the polygon in the process. Topology cleaning also removes any hanging points within the polygon thus making the polygon to hold only the key points for the study. The distance is then supplied between sample centers in the study. There is choice to allow the polygons to overlap or to have complete containment within the selected polygon shapes/graphics.

This script is further used for grouping together areas with similar variability so that they can be assigned uniform/similar number of random points during sampling.

An example is shown below of a sample script for converting histograms into polygons.

```
{
theViewList = list.Make
theDocs = av.GetProject.GetDocs
for each theDoc in theDocs
if (theDoc.is(View)) then
theViewList.Add (theDoc)
end
end
theView = msgBox.choiceAsString (theViewList, "Select the view containing the themes",
"Select a View")
if (theView = nil) then
exit
end
 'prompt the user to choose the edit and source themes
theThemesList = theView.GetThemes
theEditTheme = msgBox.ChoiceAsString (theThemesList, "Select the edit theme", "Edit
Theme")
if (theEditTheme = nil) then
exit
end
theSame = true
while (theSame)
theSame = false
theSourceTheme = msgBox.ChoiceAsString (theThemesList, "Select the source theme",
"Source Theme")
if (theSourceTheme = nil) then
exit
end
if (theEditTheme = theSourceTheme) then
msgBox.error ("The source theme cannot be the same as the edit theme." + NL + "Please
choose a different source theme.", "")
theSame = true
end
end
  }
}
```

The output from the above script is as shown in Fig. 4.2 below. The entire script is shown in Appendix B.

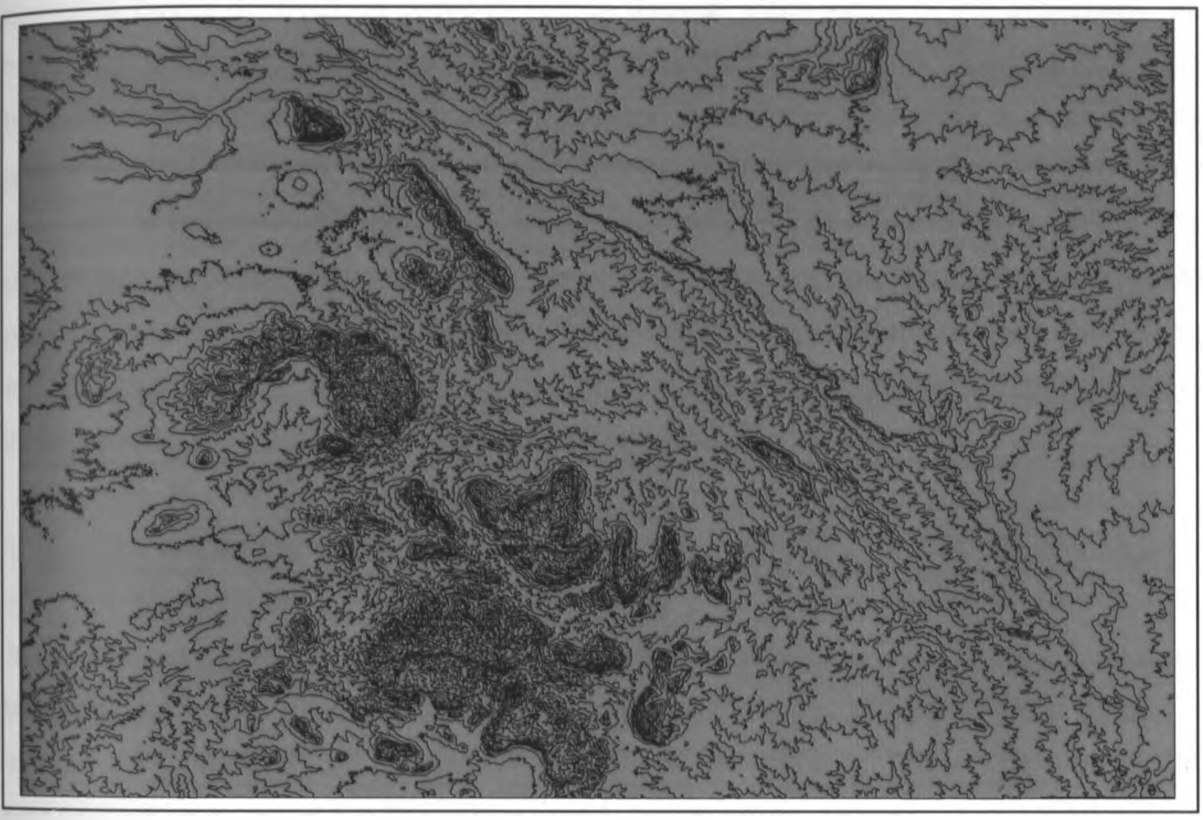


Figure 4.2: Output of histogram to polygons conversion script.

The above output shows a set of polygons based on the DEM of Upper Athi River Basin. From the above output, the neighbouring pixels were assigned the same polygons since they exhibit same spatial characteristics. Small areas and stand alone pixels were collapsed during the topology cleaning process hence,- leaving only polygons where placement of LHS samples will be done.

4.1.3 Script for random search of sampling locations

This script was written to create a set of random points within specified strata of the probability distribution obtained from the script of histogram generation as shown in Fig. 4.1. The number of strata was left open to be specified by the user (as number of sampling points desired). In order to search through each strata of the probability, the histogram generated from the above step is converted back into polygons (Fig. 4.2) so that the search is contained within the polygon boundary. This script also copies the georeferenced locations of the selected random points and uses them to produce a shapefile of randomly placed points within selected polygon features or graphics. The use of polygons was preferred since they have "hard" boundaries compared to raster pixel boundaries. A script for converting the histogram classes into polygons was written for this purpose (refer to section 4.1.2). Once established, the polygons are supplied as input for searching the optimum location for selected sample size.

This script further creates a shape file of randomly placed points within a polygon or selected polygons in an active theme. In this case, the script randomizes specific features in an active theme. For example, if there are a number of areas to be randomly selected, zones to be randomized in a given study area, etc. In random search of sampling locations of a given size, all subsets of the frame are given equal probability so that each element of the frame has equal chance of occurrence.

This script also provides the option for the systematic point sampling in random search of sampling points. It creates a systematic network of points within the polygon(s) or graphic(s) selected. The X and Y spacing and the proximity to the edge of the selected feature(s) need to be identified when using this script. It relies on arranging the target population according to some ordering scheme and then selecting elements at regular intervals through that ordered list.

An example is shown below of a sample script for random search of sampling locations.

```
{
theView = Av.GetActiveDoc
Viewgr = theView.GetGraphics
distunits = theView.GetDisplay.GetDistanceUnits
mapunits = theView.GetUnits
d = units.GetUnitString(distunits)
shapeList = {}
GROUP = FALSE     ' do we want to treat the selected graphics/polygons as a single or
seperate polygons.
tmpshp = nil

' see if there are some graphics selected. If so, generate the list of shapes to
process....
'
if (theView.GetGraphics.GetSelected.count > 0) then
  numpolys = 0
  numlines = 0
  for each agr in theView.GetGraphics.GetSelected
    if (agr.getshape.getDimension = 1) then
      numlines = numlines + 1
    elseif (agr.getshape.getdimension = 2) then
      numpolys = numPolys + 1
    end
  end
 }
}
```

An example output for the above script is given in Fig. 4.3 below. In this example 120 points have been randomly produced for the Upper Athi River Basin.

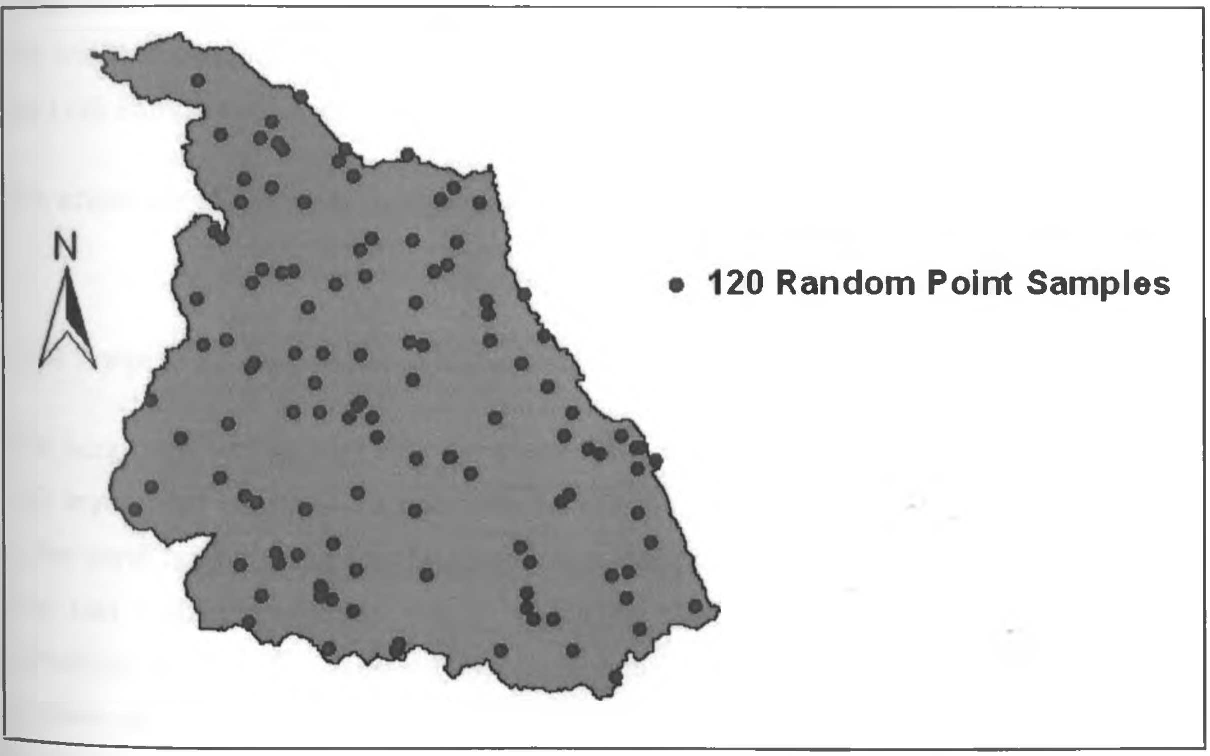


Figure 4.3: Output of random point sampling script.

From the above output, 120 points were randomly selected within the Upper Athi River Basin study area. It can be deduced from this output that the points are randomly distributed over the entire study area. The symmetric reflection of most of the samples in the study area above showed that the altitude of the sample points obtained by LHS had similar distribution as that of the entire study area. This implies that the LHS sample points contained all the characteristics of altitude as contained in the altitude for the entire study area; hence they adequately represented the elevation feature space for the Upper Athi River Basin.

Furthermore, the spread of samples points from LHS sampling method was even and uniformly spread in the study area. From this sampling characteristic using LHS as the sampling protocol tool, it can be deduced that it is the most appropriate sampling tool since it has the ability to select samples that represent a hypercube of the original data in the LHS algorithm and therefore enabling the user to build a model to predict environmental classes or attributes. A good sampling framework therefore, should be the one that samples a feature space exhaustively within the study area as depicted by the LHS sampling protocol as shown in the output above.

The entire script is shown in Appendix C.

4.1.4 Script to convert vector to raster data

This script was written for LHS cases where users have some LHS constraints as vector GIS layers, for example, in choosing certain polygons in a land use map or using a buffer zone from a given spot/line/road. The input for this script is a polygon vector (e.g. land use map, buffer zone, etc) in which the *Attribute table* has a field/column with numerical entry to be used as the pixel output value in the final raster map. The features of the script include the provision for the user to specify the output cell size and coverage.

This script is further used to convert vector to raster and to determine spatial variations of the input constraints explicitly. This can only be done on pixel basis; which is available only with raster datasets. Histogram generation, which is a core step, is only possible with raster datasets. This is because raster datasets record a value for all points in the area covered which may require more storage space than representing data in a vector format that can store data only where needed. Moreover, raster datasets allows easy implementation of overlay operations, which are more difficult with vector data.

In order to make input datasets in uniform formats, the script helps in determining the extent to which the spatial variability based on input constraints available. In GIS, for example, geographical features are often expressed as vectors, by considering those features as geometrical shapes. Different geographical features are expressed by different types of geometry. For example; Points - Zero-dimensional points are used for geographical features that can best be expressed by a single point reference; in other words, simple location.

An example is shown below of a sample script for converting vector to raster.

```
{
theView    = av.GetActiveDoc
thmThemeIn = theView.GetActiveThemes.Get(0)

' Specify the output shapefile...

  fnDefault = FileName.Make("c:\tmp\").MakeTmp("shape","shp")
  fnOutput  = FileDialog.Put( fnDefault,"*.shp","Output Shape File" )
  if (fnOutput = nil) then exit end
  fnOutput.SetExtension("shp")
  ftbOutput = FTab.MakeNew( fnOutput, POINT )
  ftbOutput.AddFields({Field.Make("ID", #FIELD_LONG, 8, 0)})
  ftbOutput.AddFields({Field.Make("New-ID", #FIELD_LONG, 8, 0)})
  ftbOutput.AddFields({Field.Make("X-Coord", #FIELD_DECIMAL, 18, 5)})
  ftbOutput.AddFields({Field.Make("Y-Coord", #FIELD_DECIMAL, 18, 5)})

' Check if having selection

  if (thmThemeIn.GetFTab.GetSelection.Count > 0) then
      colToProcess = thmThemeIn.GetFTab.GetSelection
      nRecs = colToProcess.Count
    else
      colToProcess = thmThemeIn.GetFTab
      nRecs = colToProcess.GetNumRecords
  end

'Get a List of Fieldnames that can be used
  aFields = {}
   for each f in thmThemeIn.GetFtab.GetFields
```

```
    if ( f.IsTypeNumber ) then
       aFields.Add(f)
     end
 }
```

The output of the above script is as shown Fig. 4.4 below.

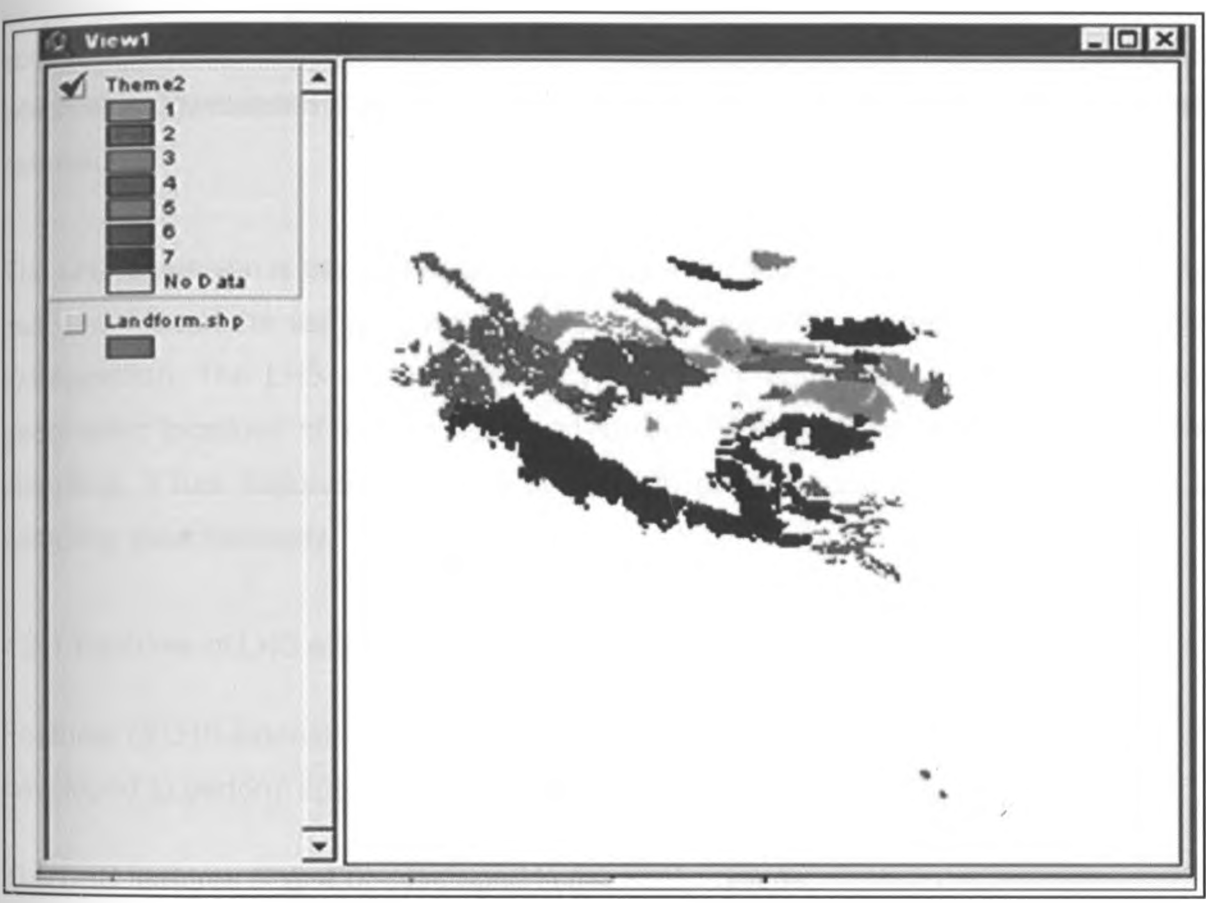


Figure 4.4: Output of vector to raster script conversion.

From the output above, it can be deduced that the vector features can be made to respect spatial integrity through the application of topology rules such as 'polygons must not overlap'. Vector data can also be used to represent continuously varying phenomena.

The entire vector to raster script is shown in Appendix D.

4.2 LHS extension in ArcView

LHS extension was developed in this study by incorporating features that provide random sampling within a bounded area as well as for selecting polygons within a boundary. It incorporates the ArcView scripts described in section 4.1 above. These scripts were integrated into one unit using the ***extbuild.apr script*** (refer to Appendix E), which was downloaded from www.esri.com on 6th December 2009 to produce the LHS extension.

The LHS extension is capable of statistically representing study area characteristics but can also spread the sampling points and in other cases they can be utilized in spatial extrapolation. The LHS extension in ArcView is fully equipped with GIS facilities for geographic locations of sample sites and also contains statistical routines for optimal sampling. It has features which can be used to perform various operations during a sampling exercise/study.

4.2.1 Features of LHS extension in ArcView

Features of LHS extension in ArcView are the functions/codes that are designed and developed to perform specific functions and operations in a sampling exercise/study.

The LHS features (Fig. 4.5) developed in this study include;

a) Random point sampling feature and
b) Systematic point sampling feature

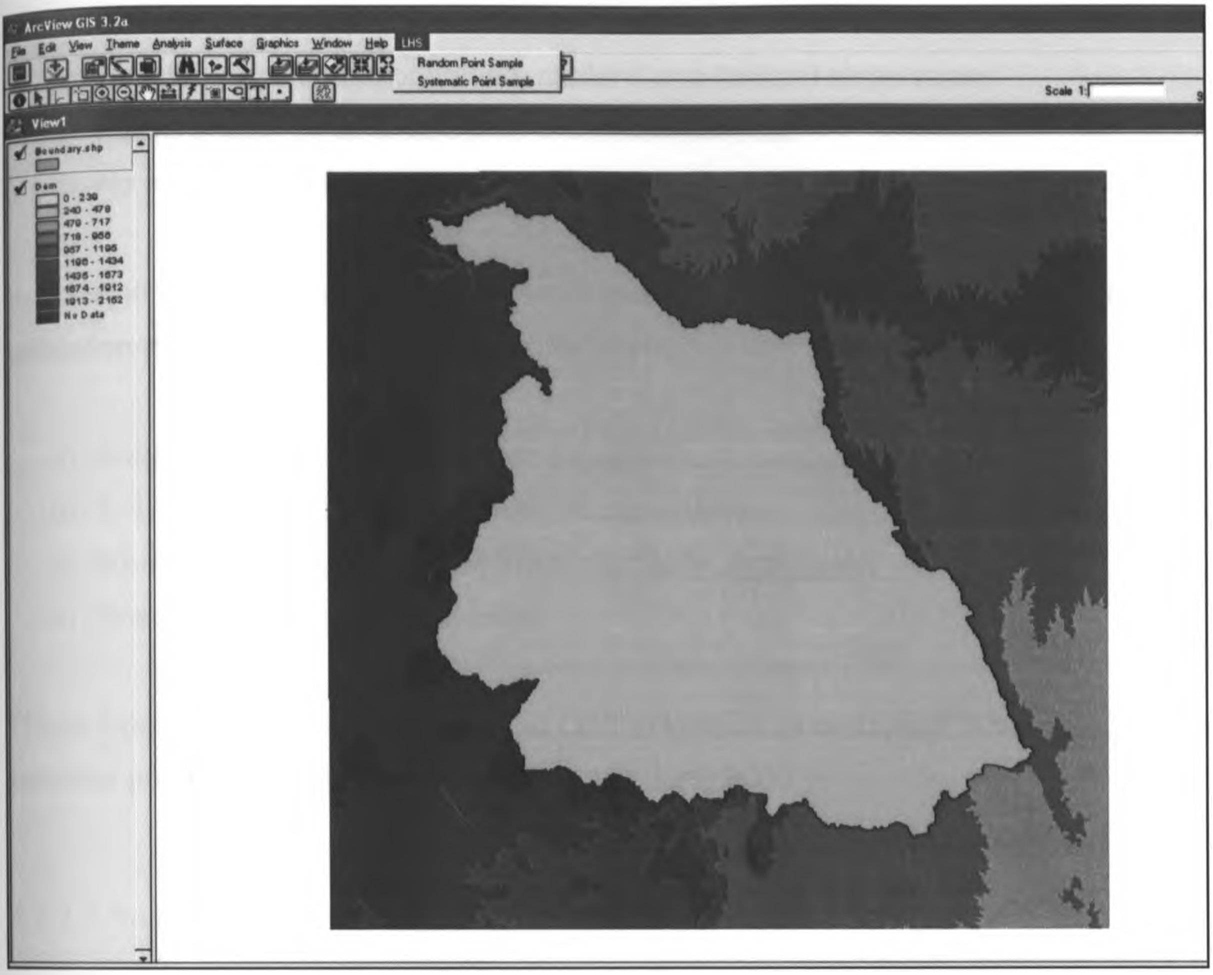


Figure 4.5: Window showing LHS extension features in ArcView.

The above sampling features were designed to assist environmental engineers and/or field scientists in using ArcView to generate spatially explicit random and systematic sampling protocols to support resource monitoring, mapping and research needs. These features work either with polygons in a theme or with graphics that have been added to a View. Samples can be entirely within a single polygon or shape, or distributed among several disjunct polygons or shapes. Possibilities for a number of user-defined constraints and settings are also offered as input options. The following constraints guided the choosing of the validation points in this study;

a) 20 random points to be selected within the study area
b) Equal representation of upland, midland and lowland slope zones
c) Points should be at least 500 m from the main road
d) No point should fall in urban centres and water bodies

In addition, the following GIS inputs were used to implement the LHS selection of the validation points;

a) Boundary of the study area (see Fig. 3.13)
b) 1: 100,000 Land cover map from Omuto (2008)
c) 90-m spatial resolution DEM (from http://srtm.usgs.gov)
d) Road network for the study area

These inputs were processed using the LHS extension as described in section 3.2. The selected points were visited in the field on 15th June 2009.

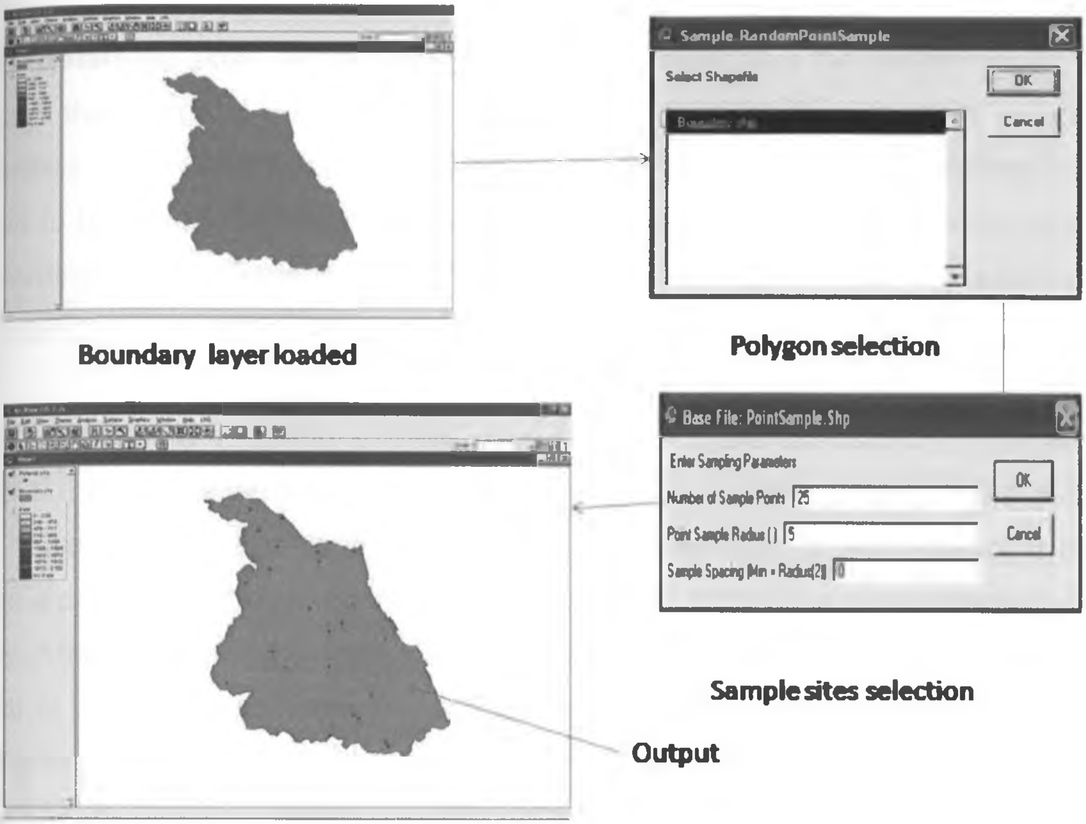
4.2.1.1 Random point sampling feature

This feature creates a shape file of randomly placed points within a polygon or selected polygons in an active theme. It can also be used to randomize specific features in an active theme, for example, if there are a number of areas to be randomly selected, zones to be randomized in a given study area, etc. In random point sampling of a given size, all subsets of the frame are given equal probability so that each element of the frame has equal chance of occurrence.

Random point sampling is the best form of probability sampling. This is because each member/sample of the population has an equal and known chance of being selected. When there are very large populations, it is often difficult to identify every member/sample of the population; hence the pool of available subjects becomes biased. Random point sampling is obtained by choosing elementary units in such a way that each unit in the population has an equal chance of being selected. Generally, random point sampling is free from sampling bias.

However, using a random number table to choose the elementary units can be cumbersome. For example, if the sample is to be collected by a person untrained in statistics, then instructions may be misinterpreted and selections may be made improperly. Random point sampling in a plane may be used as sampling points or in connection with modeling — for example as part of a Monte Carlo simulation of a probability distribution. Monte Carlo methods are a class of computational algorithms that rely on repeated random sampling to compute their results. Monte Carlo methods are often used in simulating physical and mathematical systems. Because of their reliance on repeated computation of random or pseudo-random numbers, these methods are most suited to calculation by a computer and tend to be used when it is unfeasible or impossible to compute an exact result with a deterministic algorithm.

The following steps shown in the Fig. 4.6 below were followed. The boundary study area was loaded and polygon selection was carried out. The number of samples was selected in a random manner based on a sample radius and sample spacing. The output is as shown below.



Select Shapefile

Boundary.shp

Enter Sampling Parameters

Number of Sample Points 25

Point Sample Radius () 5

Sample Spacing (Min = Radius(2)) 10

Figure 4.6: Random point sampling in LHS.

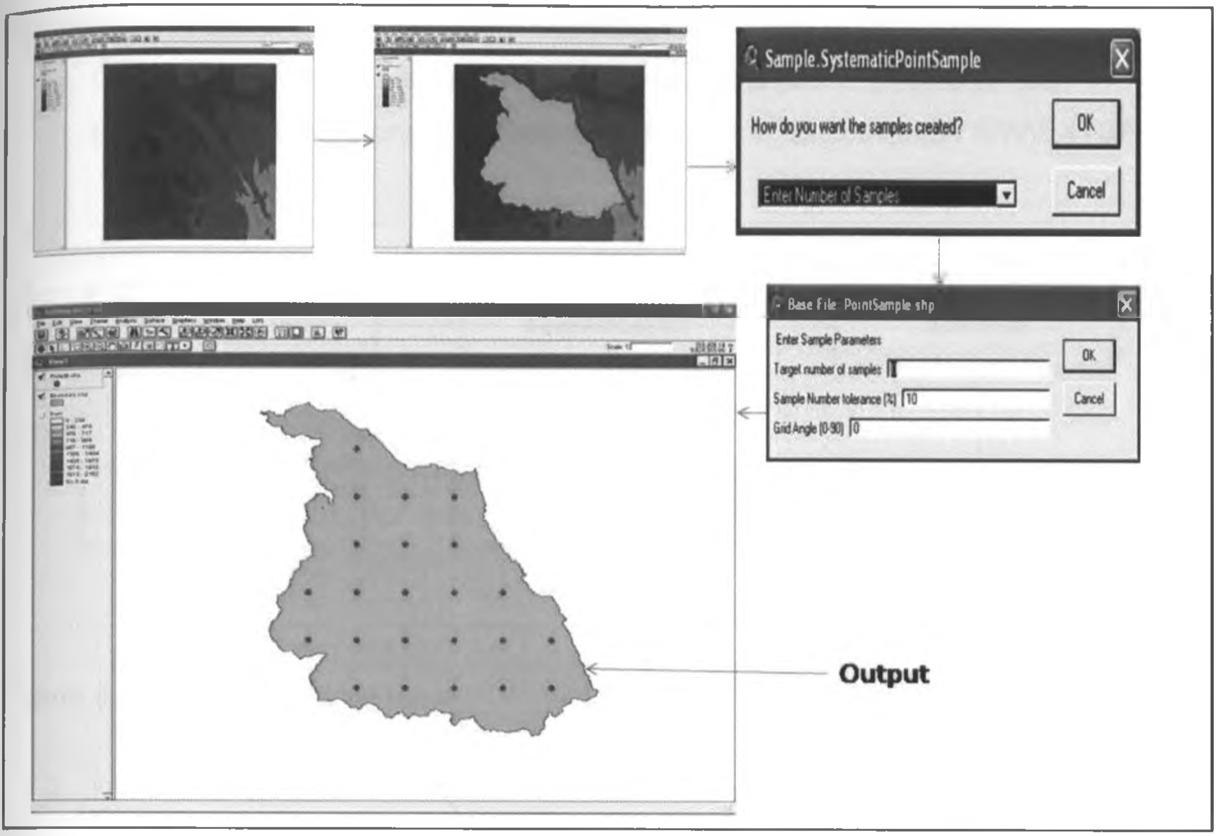
From the above output, 25 samples were selected. It can be seen that these samples are randomly distributed all over the study area. Though the samples are randomly placed, they are evenly distributed, that is, no samples are clustered in one place.

4.2.1.2 Systematic point sampling feature

The systematic point sample option works very much like the random point sample except that it creates a systematic network of points within the polygon(s) or graphic(s) selected. The X and Y spacing and the proximity to the edge of the selected feature(s) need to be identified when using this option. It relies on arranging the target population according to some ordering scheme and then selecting elements at regular intervals through that ordered list.

Systematic point sampling is often used instead of random sampling (Kelsey et al., 1986). It is also called an nth name selection technique (Kelsey et al., 1986). It is called nth name selection technique because during the selection of numbers/points/samples from a population, the researcher would only picked samples at some specified intervals only. After the required sample size has been chosen, every nth record is selected from a list of population members. As long as the list does not contain any hidden order, this sampling method is as good as the random sampling method. Its only advantage over the random sampling technique is simplicity.

The following steps shown in the Fig. 4.7 below were followed. The boundary study area was loaded and polygon selection was done. The number of samples was selected in a systematic manner based on a sample radius and sample spacing. The output is as shown below.



Sample.SystematicPointSample

How do you want the samples created?

Enter Number of Samples

Base File: PointSample.shp

Enter Sample Parameters

Target number of samples:

Sample Number tolerance (%) 10

Grid Angle (0-90) 0

Figure 4.7: Systematic point sampling in LHS.

From the above output, 23 samples were selected. It can be seen that these samples are systematically distributed all over the study area. The samples are evenly distributed across the feature space/study area.

In order to use the LHS extension, the following steps need to be followed:

1. Copy **LHS.avx** file to ArcView's 32-bit extension directory. On Windows platforms, this directory is located at; C:\ESRI\AV_GIS30\ARCVIEW\EXT32.

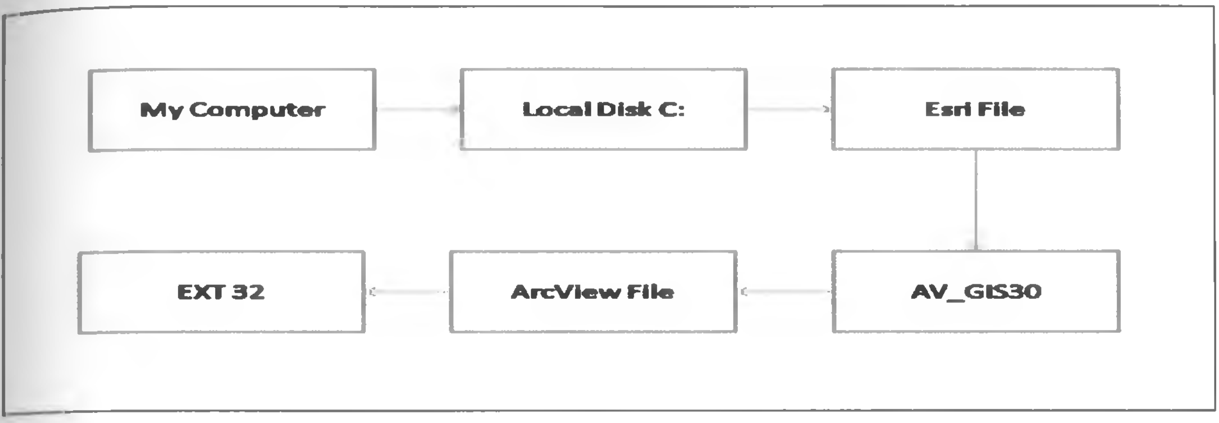


Figure 4.8: Figure showing the location for LHS Extension (LHS.avx) in a computer.

2. Load LHS extension onto ArcView GUI. This can be loaded from ArcView's File menu by selecting the Extensions menu choice then LHS. After it is loaded, a menu labeled LHS is added to the GUI panels as shown.

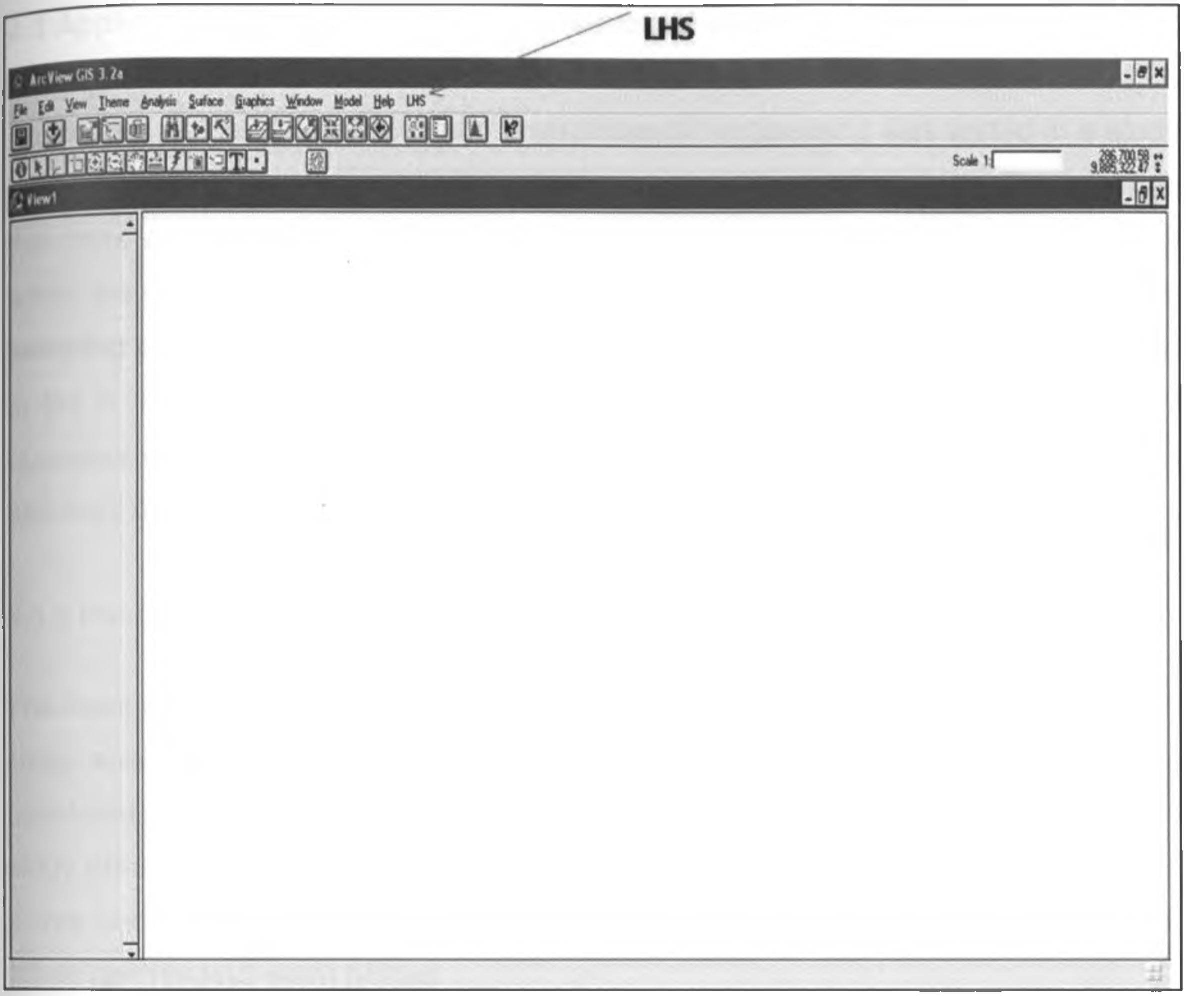

LHS
ArcView GIS 3.2a
File Edit View Theme Analysis Surface Graphics Window Model Help LHS
Scale 1:
286,700.58
9,885,322.47
View1

Figure 4.9: Window showing LHS extension in ArcView.

4.3 Application of LHS in typical environmental engineering problems

Application of the LHS extension in environmental engineering was tested in a study of loss of topsoil in the Upper Athi River Basin in upper Eastern province. The constraints that were considered for the sampling study included: 20 LHS random points selected within the study area; equal representation of upland, midland and lowland slope zones; sampling points were at least 500 m from the main road and no point was to be allowed to fall in urban centres and water bodies. These constraints were obtained from; the boundary of the study area, 1: 100,000 land cover map, 90-m spatial resolution DEM and road network for the study area.

4.3.1 Preparation of the input constraints into GIS data layers

The input constraints that were considered for GIS data layers included the DEM of the study area which was obtained from a 90-m spatial resolution of DEM. Other input constraints that were considered included 20 LHS random points selected within the study area; equal representation of upland, midland and lowland slope zones; sampling points were at least 500 m from the main road and no point was to be allowed to fall in urban centres and water bodies.

The DEM of the study area was uploaded in ArcView (Fig. 4.10) and then reclassified into three (3) classes namely upland, midland and lowland slope zones using spatial analyst extension in Arcview (Fig. 4.11).

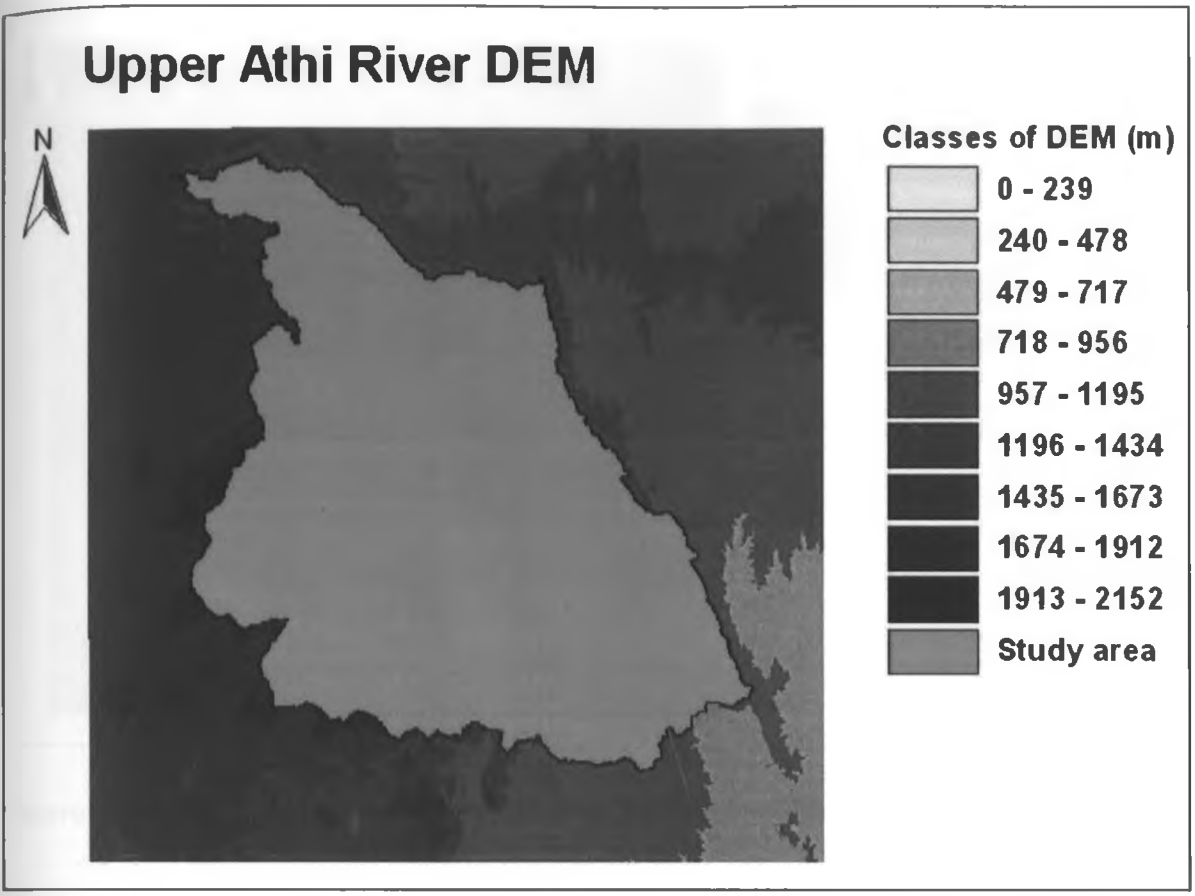


Figure 4.10: DEM of Upper Athi River Basin.

The DEM of Upper Athi River Basin shown above (Fig. 4.10) depicts in detail the variations in elevations of the area. It also shows the general terrain/landscape of the area under study. DEM's plays also a key role in spatial dataset required for catchment and water resource management. Furthermore, DEM's can be used in various applications including natural resource management, flood assessment and management, asset protection and land planning. Other applications of DEM's include topographic mapping, view shed analysis and soil erosion monitoring.

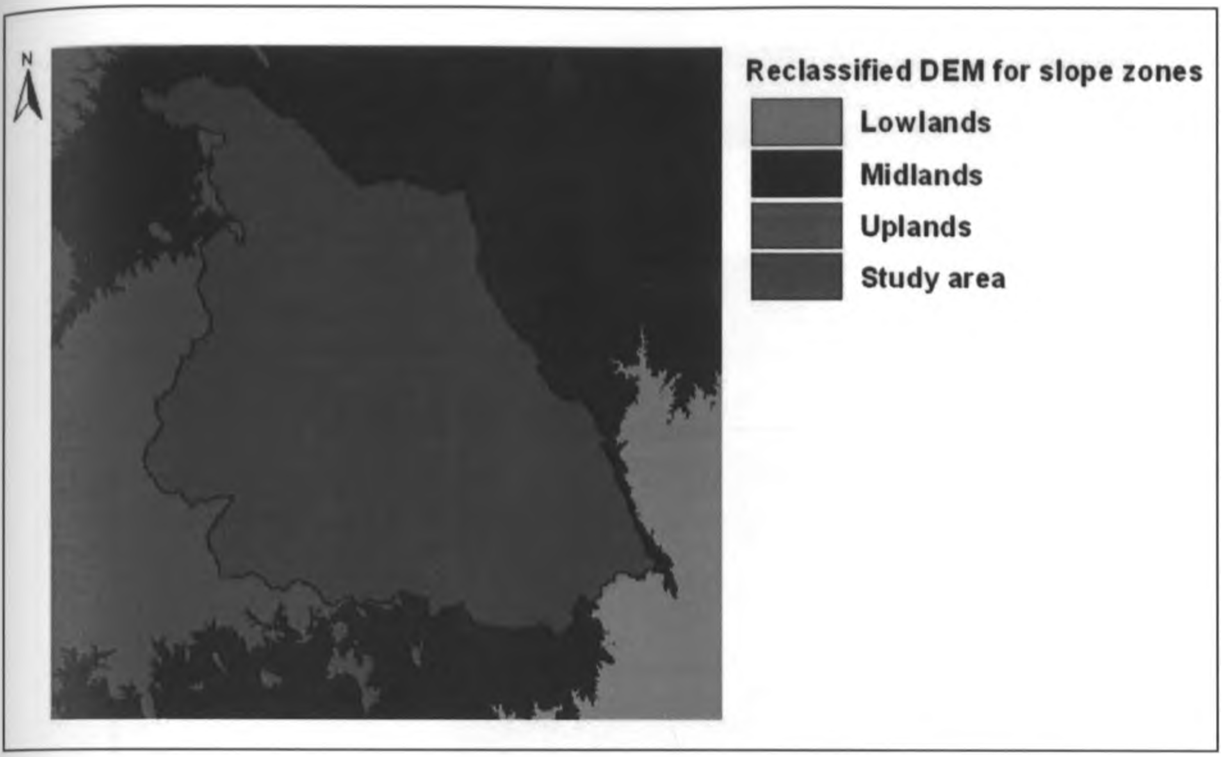


Figure 4.11: DEM classes showing Upland, Midland and Lowland slope zones.

The DEM classes shown above (i.e. Upland, Midland and Lowland slope zones) describe in detail the topography of the area under study. It shows that Upper Athi River Basin comprises of areas with high elevations, other areas with average elevations and some areas with low elevations. The classes shown above too describe further in detail what economic activities (i.e. agricultural or farming) are practiced in high, medium and low altitudes areas of the study area. This reclassification also depicts/shows the climate variations in the area based on rainfall intensity in the three zones.

The DEM of the study area was then converted into a histogram (Fig. 4.12). The histogram shows in detail which parts of the study area have high altitude/elevations compared to others. The histogram helps in indentifying out low, medium and high elevations within the topographical area under study.

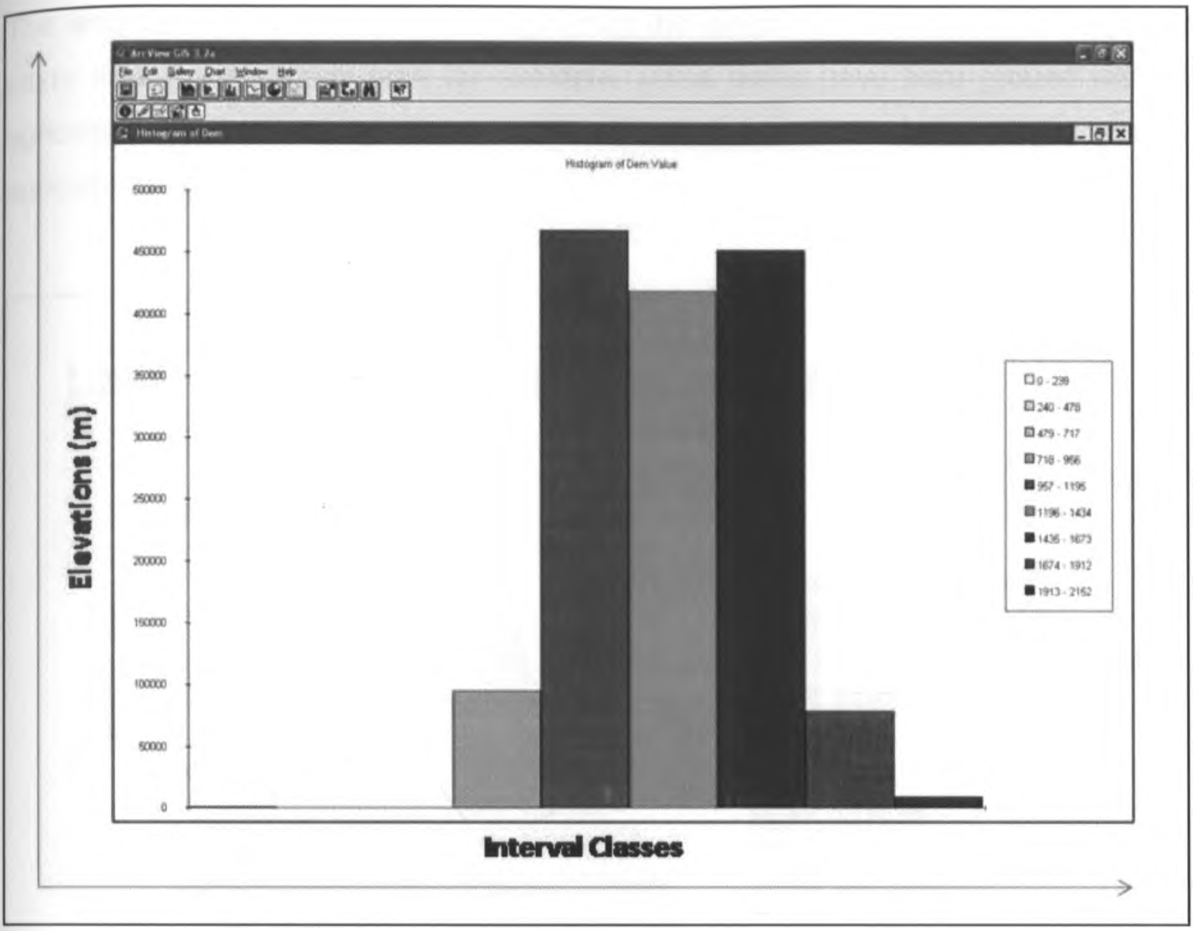


Figure 4.12: Histogram of DEM for Upper Athi River Basin.

The land use (LU) map showing various economic occupations taking place in study area was also loaded into ArcView. Its polygons were reclassified into numeric units and the final output converted to a raster format (Fig. 4.13). The land use map describes in totality the major and minor economic activities of the people in the study area. For example, from the land use map, the major economic activity in the study area is agriculture. People in Upper Athi River Basin depend majorly on agricultural activities to fend for themselves and their families. Heavy agricultural activities within the area have led to heavy loss of rich top soil by erosion. Intense farming in an area without regard to good agricultural practices causes loose soils to be swept away either by rain water or wind when the ground is bare.

The land use map also helps in differentiating the varied land cover available within the study area. In this study area for example, some areas have bare ground with less agricultural activity taking place while others dense bush land (virgin) with no agricultural activity taking place at all.

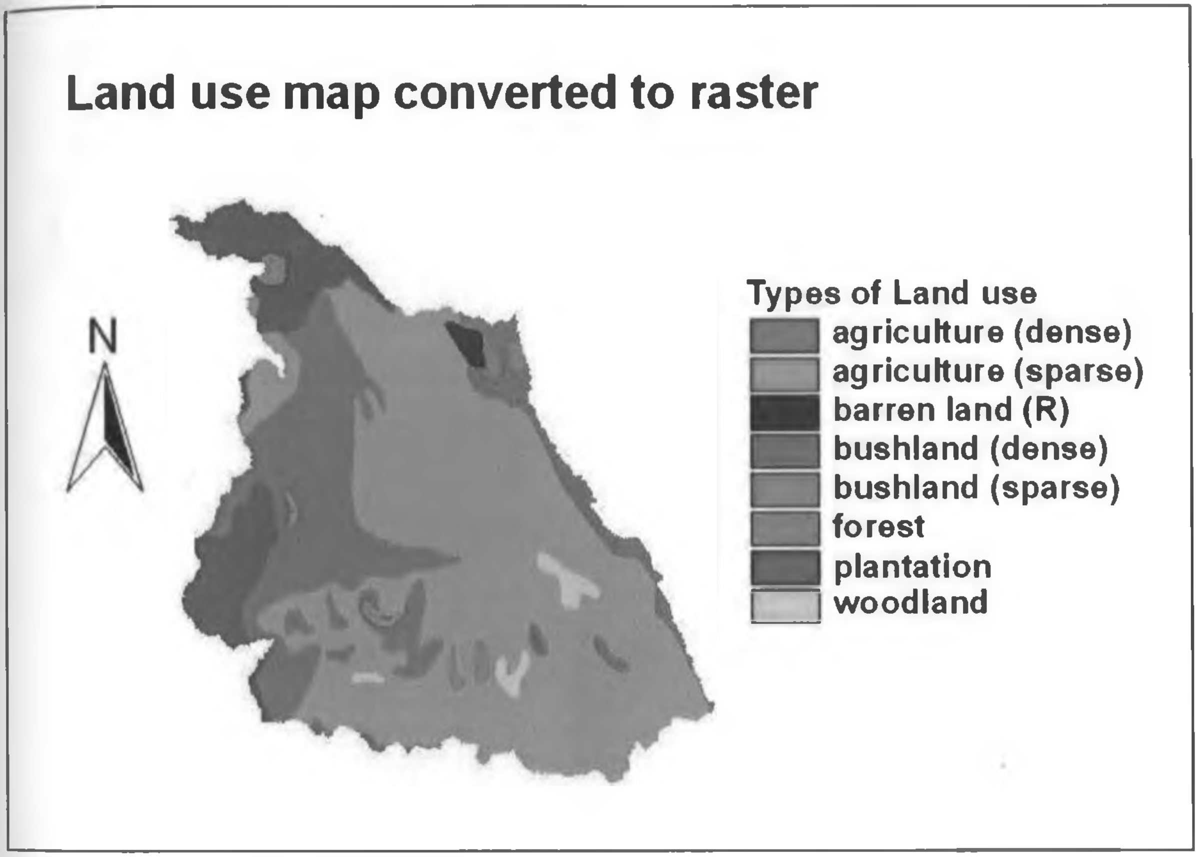


Figure 4.13: Land use map converted to raster.

The land use map was useful as a constraint because from it, various land cover and varied agricultural activities can be identified.

The road network map for the study area was also loaded in ArcView as shown in Fig. 4.14 below. The road network in the study area depicts the extent to which the communication system in the area has been developed. It shows whether the area under study is accessible or not. Some areas are more accessible than others because they have got more roads. This is attributed to the fact that there are varied economic activities within the area. Areas with more economic activities, for example, agricultural activities tend to have more road network facilities than the areas with less or no economic activities taking place at all.

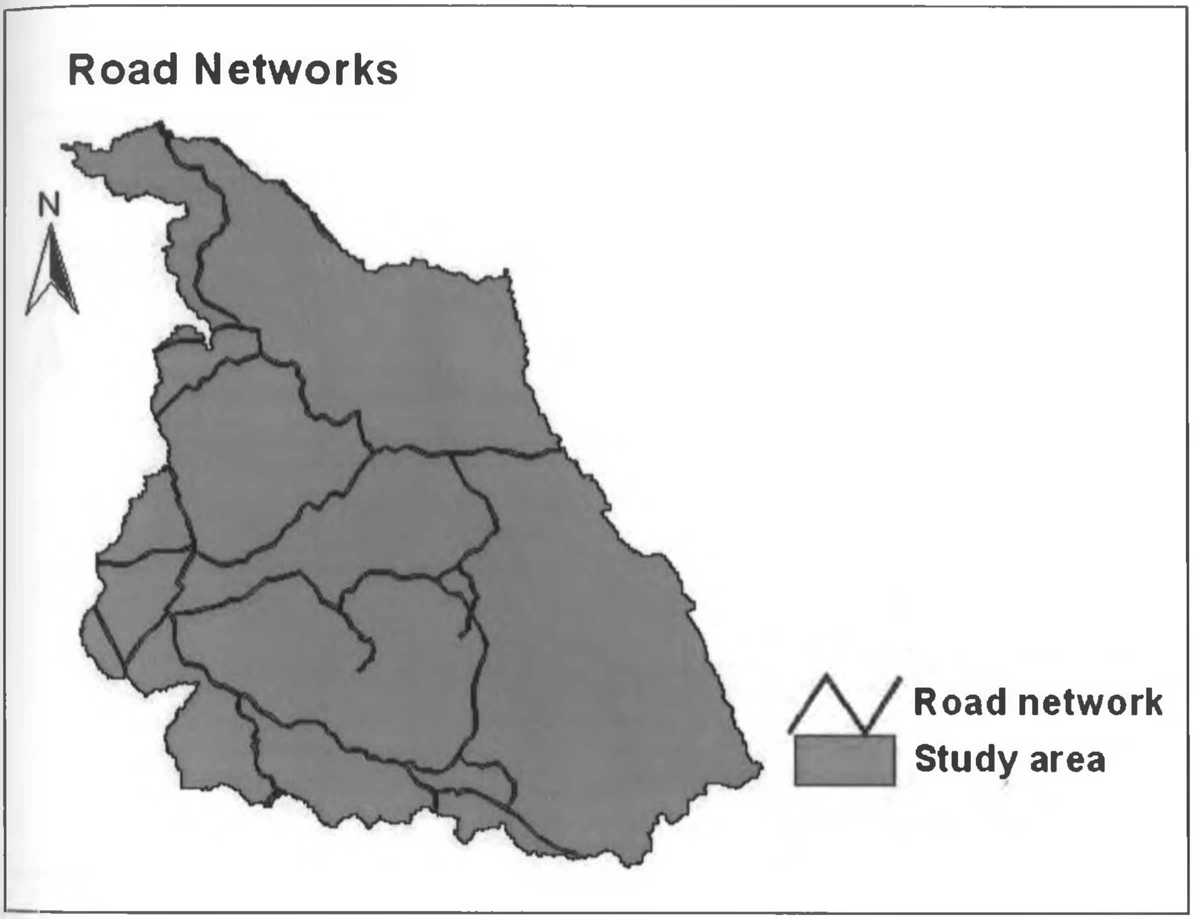


Figure 4.14: Map showing road network in Upper Athi River Basin.

A 500 m wide buffer was created in all the roads within the study area (Fig. 4.15). The 500 m wide buffer zone is an area where during the random sampling process; no sample should fall within this area. It is a constraint that has been factored in during the sampling process in this study. The 500 m wide buffer zone was then converted to raster format. The choice of 500 m buffer was based on the following reasons:

a) The sampling points should not be close to the road for convenience purposes during the sampling process i.e. less or no interference from human developments or activities.

b) A 500m buffer zone ensures that the samples are not clustered in one area (good sampling means that the samples are widely and evenly spread within the study area).

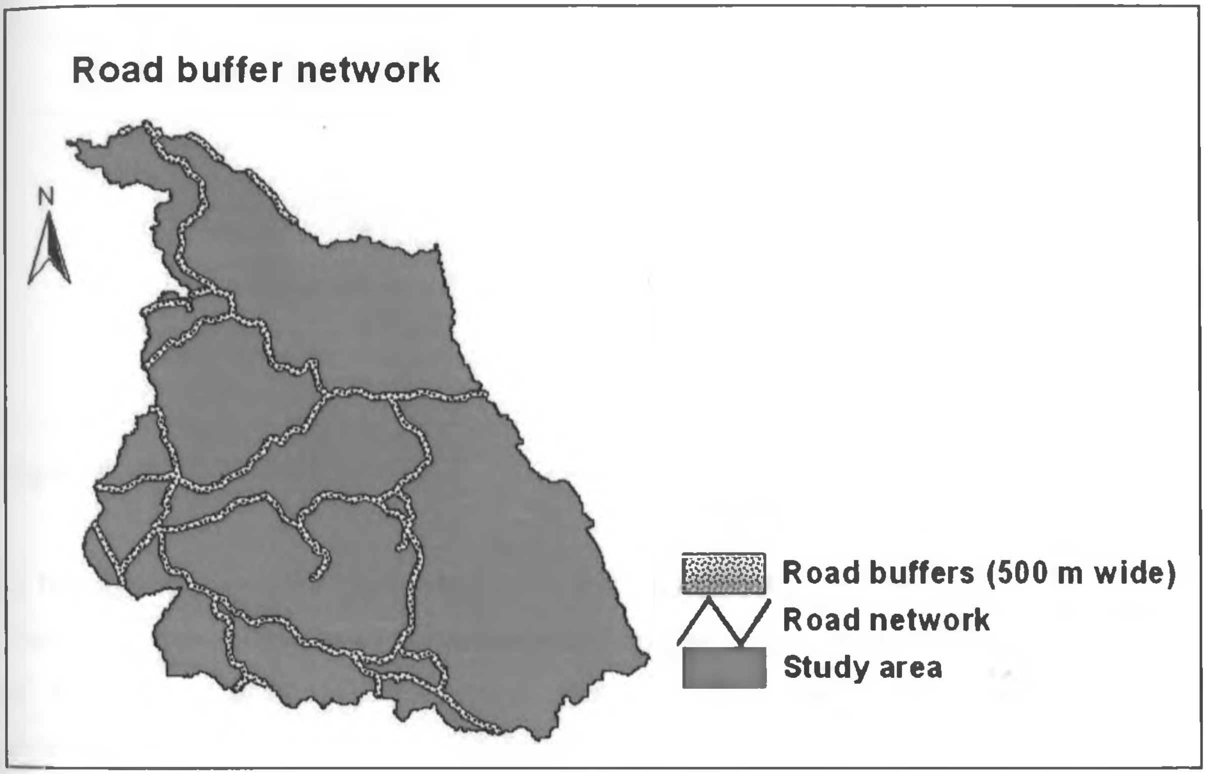


Figure 4.15: Map showing road buffer zone (500 m wide).

4.3.2 Application of LHS in environmental engineering problems

After all the three inputs were converted to rasters i.e. reclassified DEM, land use map and road network map, the areas required for LHS sampling were then chosen. Great care and proper sampling criteria were taken into consideration during the sampling process. The final outputs i.e. the weighted average of the three inputs were then combined into one single raster in ArcView. The output is as shown in Fig. 4.16.

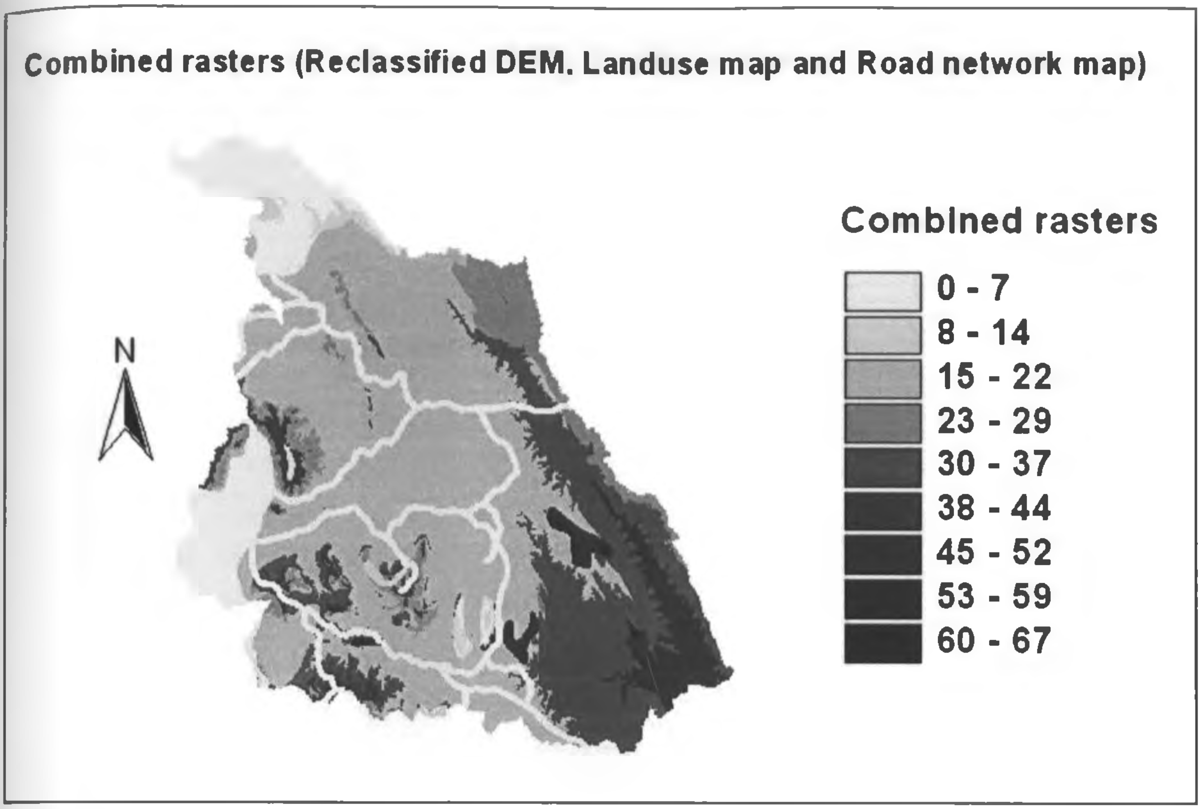


Figure 4.16: Combined rasters.

A histogram was then generated from the combined raster as shown in Fig. 4.17. Histogram generation is an important step because it shows the probability distribution of the active layer. As discussed earlier in Fig. 4.12, a histogram helps in comprehending and identifying out the variations in elevations/heights within the study area. It further shows the sampling distribution within the study area.

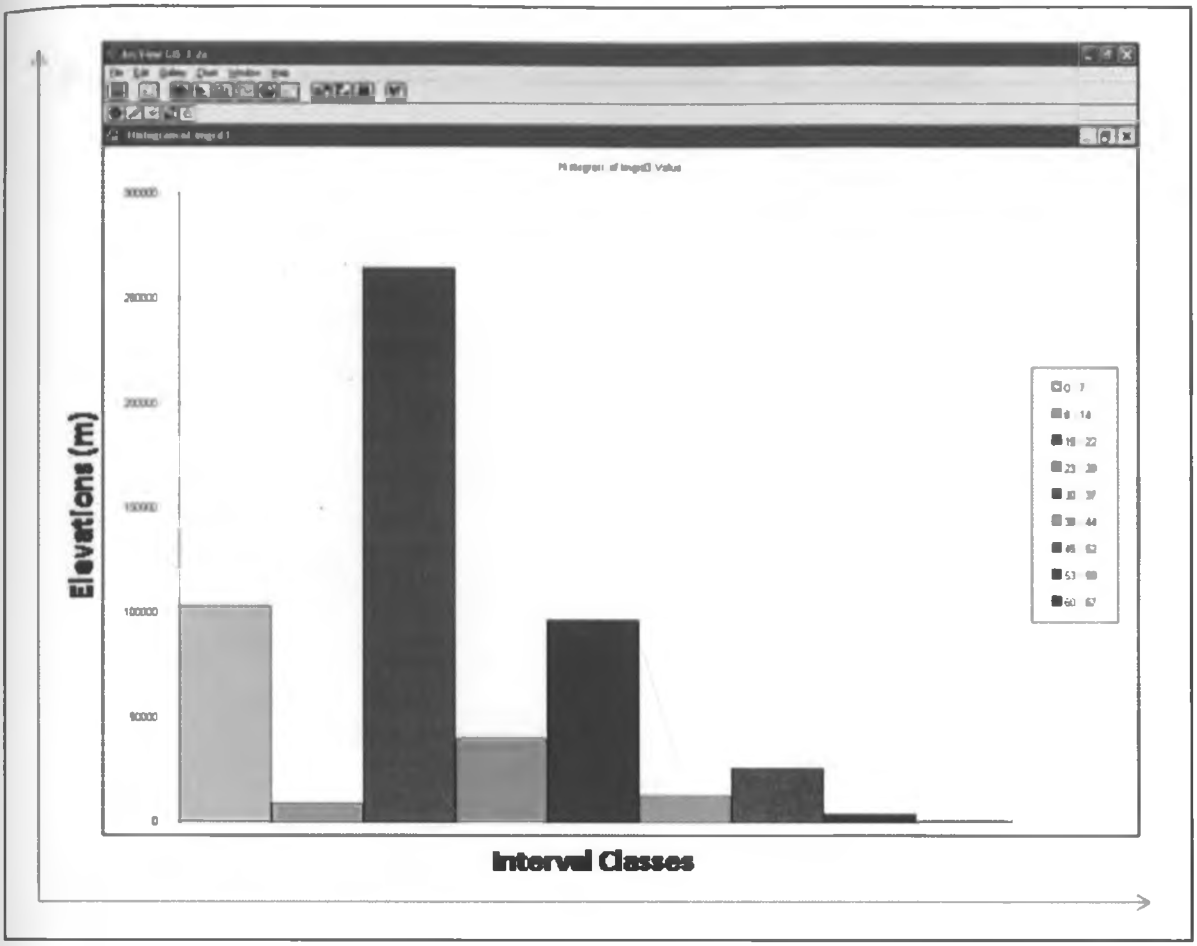


Figure 4.17: Histogram of combined rasters.

After the histogram of combined rasters was displayed as shown in Fig. 4.17, they were then reclassified into 20 classes as shown in Fig. 4.18. Reclassification of the DEM into classes provides/shows a good view of how the study area is. It provides a glimpse on the spatial variability within the study area based on elevations/heights and the various economic activities being undertaken in the study area.

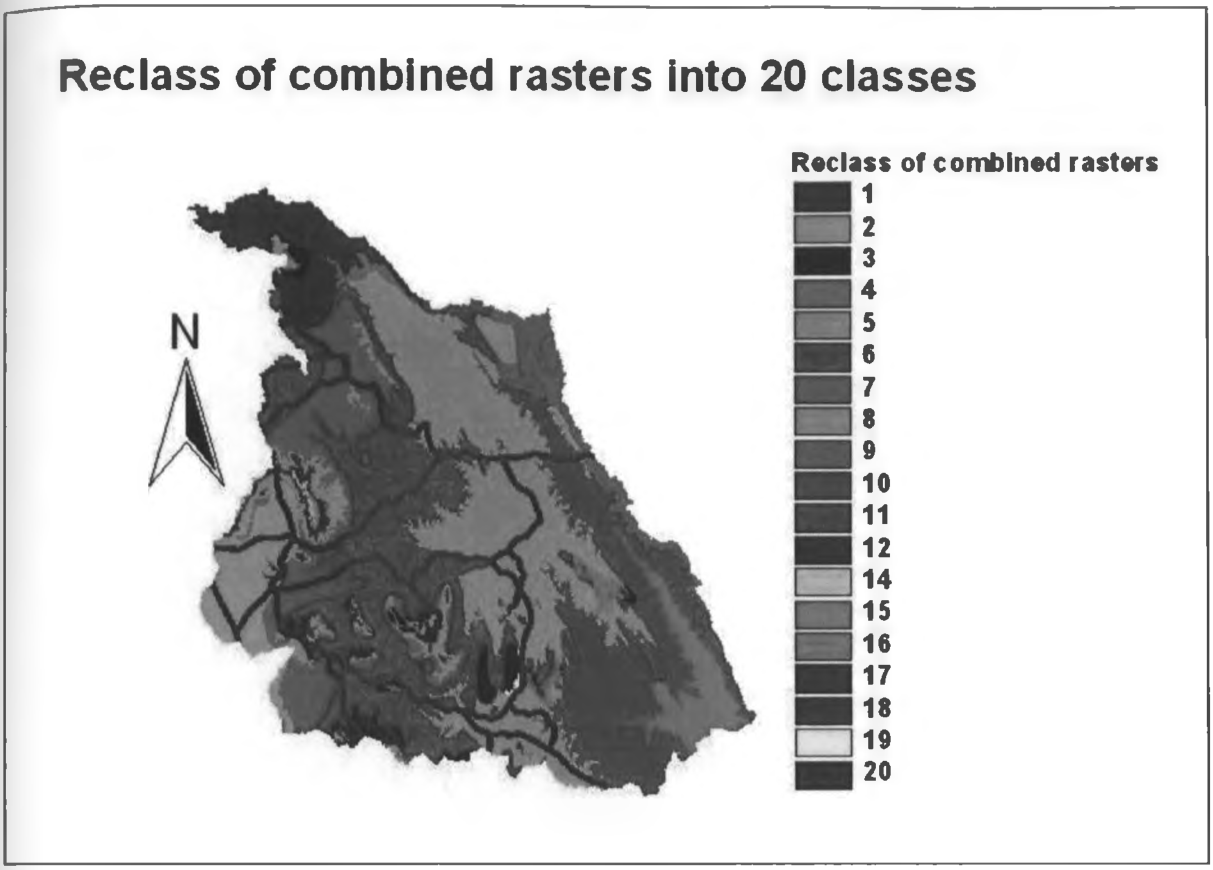

Reclass of combined rasters into 20 classes
N
Reclass of combined rasters
1
2
3
4
5
6
7
8
9
10
11
12
14
15
16
17
18
19
20

Figure 4.18: Combined rasters reclassified into 20 classes.

The histogram shown in Fig. 4.17 was then standardized and its variance established. The variance was used in VQT to obtain the required bins for generating twenty (20) random points. Twenty (20) randomly selected points were chosen for the resultant map in two ways: Systematic and Random sampling. The 20 samples picked for this study are as shown in Fig. 4.19 below.

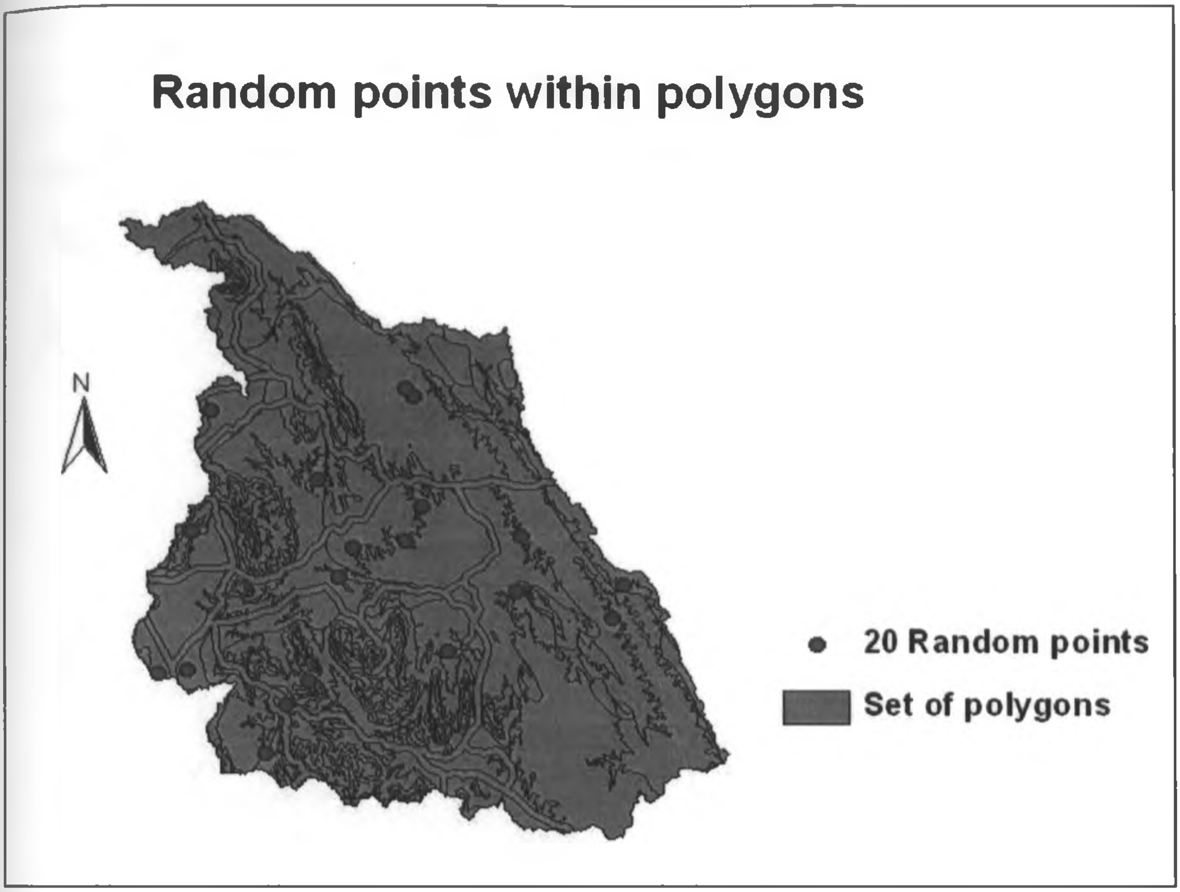


Figure 4.19: Random points within polygons.

Fig. 4.20 below shows 20 random points selected which fell outside the buffer zones.

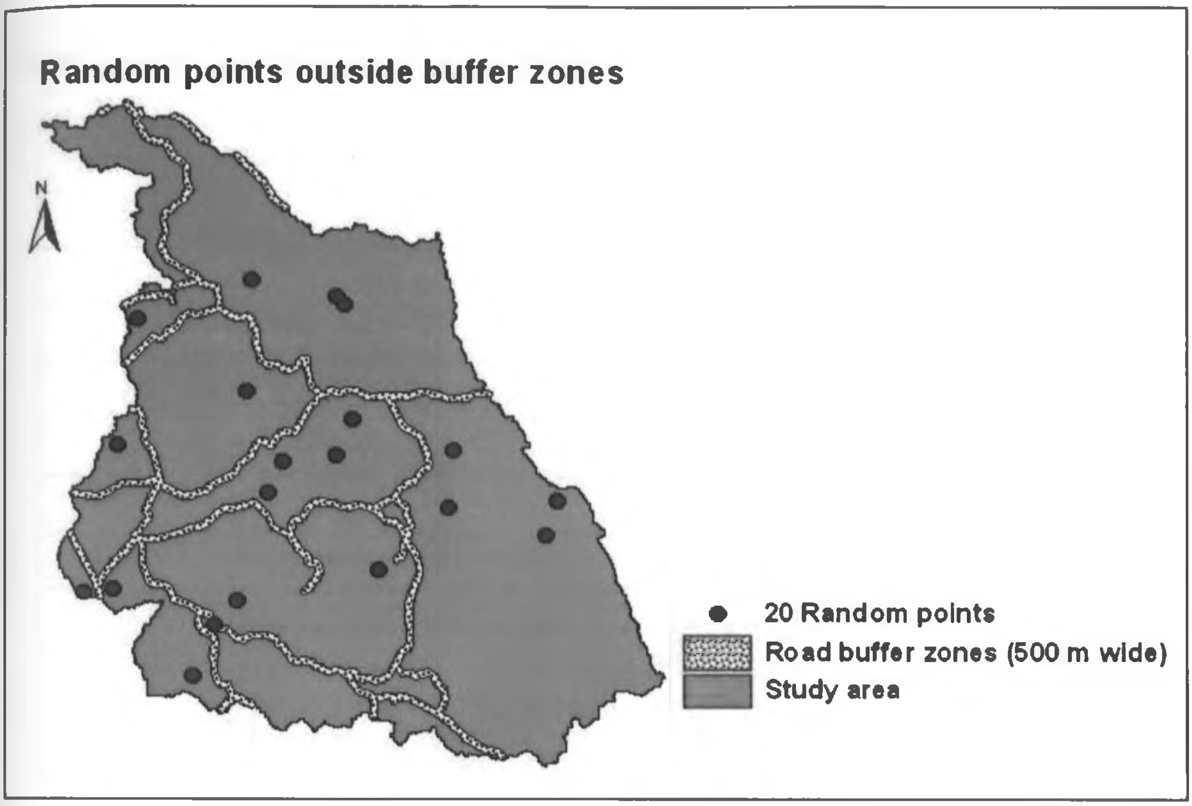

Random points outside buffer zones
N
● 20 Random points
Road buffer zones (500 m wide)
Study area

Figure 4.20: Random points outside buffer zones.

These points were then visited in the field and topsoil loss measured. The results of soil erosion are tabulated below (Table 1) and its' scatter diagram is as shown in Fig. 4.21.

Table 1: Measured and calculated soil loss in ton/ha/yr.

Sample	Measured soil loss (tonnes/ha/yr)	Calculated soil loss (tonnes/ha/yr)
1	2.80	1.90
2	0.20	2.36
3	2.80	2.99
4	6.20	12.45
5	5.50	12.40
6	3.90	8.53
7	8.30	13.02
8	1.80	4.74
9	4.50	5.90
10	9.60	13.22
11	7.30	9.20
12	5.80	8.99
13	5.00	7.24
14	9.50	11.48
15	4.70	5.02
16	5.00	8.26
17	1.30	1.55
18	4.00	9.95
19	3.10	5.46
20	6.90	7.56
Total	98.20	152.20
Average	4.91	7.61

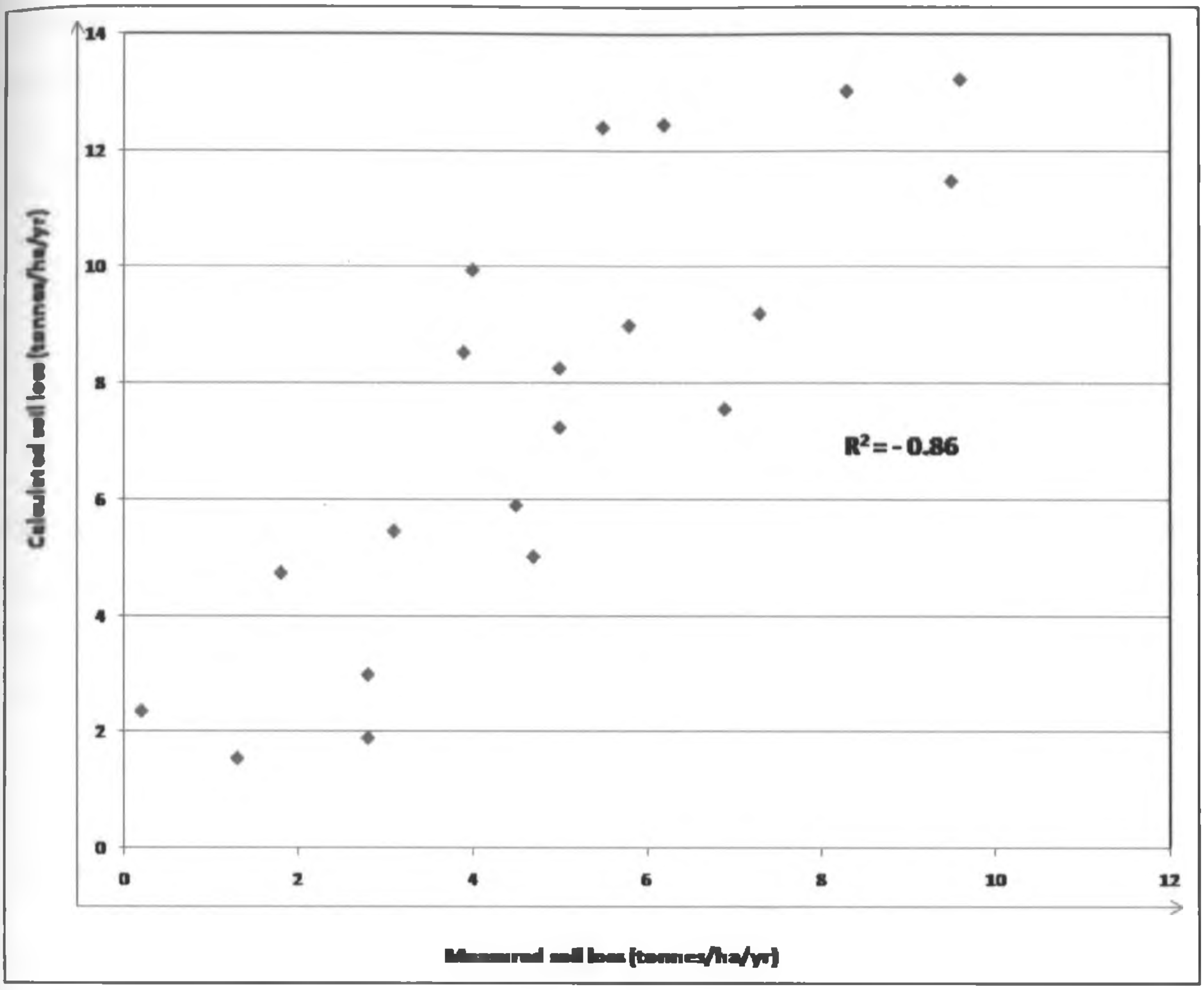


Figure 4.21: Scatter diagram showing comparison between measured and calculated soil loss (tonnes/ha/yr).

From the above tabulated results, the coefficient of efficiency can thus be calculated as follows (reference equation 3.8 in Chapter 3), that is;

$$R^2 = 1 - \frac{\sum_{i=1}^{20}(L_{vi} - L_{0i})^2}{\sum_{i=1}^{20}(L_o - \overline{L_o})^2}$$

$$= 1 - \frac{234.532}{126.178}$$

$R^2 = - 0.8587$

From the above calculations, R^2 = - 0.86. From this result, the following observations can be deduced;

a) The low values of the measured soil loss compares with the low values of the calculated soil loss throughout the study area.
b) The high values of the measured soil loss compares with the high values of the calculated soil loss throughout the study area.
c) The negative (–ve) coefficient of efficiency means that the observed mean is a better predictor than the model.

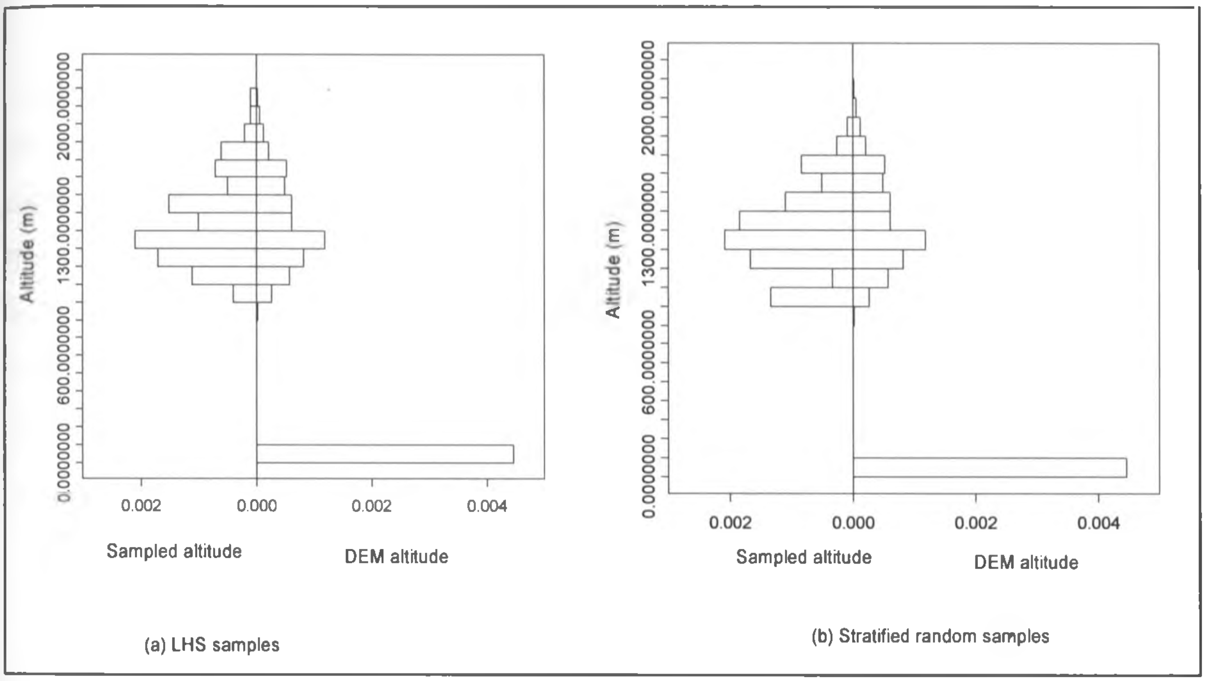
Hence, it can be deduced that LHS is capable of spreading evenly the samples within the entire study area as it can be seen from the scatter diagram (Fig. 4.21). Therefore, it can be used to validate a model. Good sampling is good for model validation since the samples are evenly distributed within the study area.

4.4 Testing of LHS extension

The performance of LHS in allocating sample points was compared to the popularly used sampling method of random sampling. In order to test the performance of the LHS extension, the sample points generated by LHS sampling frame were compared to those from a study of soil erosion in Eastern Kenya (Omuto, 2008). In this study, 120 sampling points were used to study soil erosion. These points were compared to 120 randomly selected points from the LHS extension. The comparison was done, first, on the extent to which the two sets of sampling points adequately represented the altitude feature space and then the geographic spread of the sample points in the study area.

In the comparison of the representation of the altitude feature space, back-to-back histogram comparison was used (Fig. 4.21). In this case, the back-to-back principle was used to compare sampling by LHS code and other popularly used protocols (such as random sampling and stratified random sampling). From the comparison results, the

LHS samples results mirrors the population distribution as compared to the results found from stratified random sampling. LHS extension in ArcView sampling protocol provided a good sampling cover of the environmental variables. LHS extension in ArcView enables the user to adequately sample strategic locations within the study area and ensures that a maximum area is covered well.



(a) LHS samples

(b) Stratified random samples

Figure 4.22: Back-to-back histogram results for comparing LHS & stratified random samples.

The symmetric reflection of the histogram bars in (Fig. 4.22 a) above showed that the altitude of the sample points obtained by LHS had similar distribution as that of the entire study area. This implies that the LHS sample points contained all the characteristics of altitude as contained in the altitude for the entire study area; hence they adequately represented the altitude feature space for the Upper Athi River Basin. In contrast, the unsymmetrical reflection of the histograms (Fig. 4.22 b) shows that the altitude of the sample points obtained by LHS had different distribution as that of the entire study area. This implies that the stratified sample points'did not contain all the characteristics of altitude as contained in the altitude for the entire study area; hence they did not adequately represent the altitude feature space for the Upper Athi River Basin.

Furthermore, the spread of samples points from LHS sampling method were even and uniformly spread of sample points in the study area. In contrast, the spread of the sample points obtained from stratified sampling method was uneven across the study area.

LHS therefore, was the most appropriate sampling tool. Its ability to select samples that represent a hypercube of the original data in the LHS algorithm enables the user to build a model to predict environmental classes or attributes. The Latin hypercube generally gives a good representation of the ancillary variables as shown in the back-to-back histogram criteria. A good sampling framework therefore, should be the one that samples a feature space exhaustively within the study area as depicted by the LHS sampling protocol.

5: CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

This chapter presents an overview of this study and gives a review of the important contributions and applications of Latin Hypercube Sampling (LHS) extension in a GIS environment. This has been seen in this thesis where the two objectives of this study have been met: development of avenue scripts for Latin Hypercube Sampling (LHS) method using avenue computer language and development of an add-on program in ArcView GIS for implementing the code in the first objective.

5.1.1 Development of avenue scripts for LHS method

Four avenue scripts were developed to facilitate implementation of LHS in ArcView GIS: a script for generating histograms, a script for converting the histogram bins into polygon shapefiles, a script for optimizing sampling locations within the polygons given a set number of sampling points and a script for converting vector to raster (for cases with input constraints in vector data formats). The entire codes for the scripts developed in this study are detailed in the Appendix section.

5.1.2 Use of LHS scripts to produce an add-on program in ArcView GIS

The avenue scripts developed in this study as per the first objective in section 5.1.1 above were then integrated into one unit using the ***extbuild.apr script*** to produce an LHS extension. The LHS extension was developed by incorporating features that provide random sampling within a bounded area, along a transect line, as well as for selecting polygons within a boundary. The LHS extension developed is capable of statistically representing study area characteristics but can also spread the sampling points in case they are to be utilized in spatial extrapolation. The LHS extension in ArcView is fully equipped with GIS facilities for geographic locations of sample sites and also contains statistical routines for optimal sampling. It has features which can be used to perform various operations during a sampling exercise. These capabilities qualify the extension beyond the available computer codes, which are either capable of accurate statistical determination of sampling points or geographic location of sampling sites. The main features of LHS extension that were developed and discussed in this study include: Random point sampling feature and Systematic point sampling.

5.1.3 Environmental engineering application of LHS extension

The engineering application of this study was exemplified by the soil erosion assessment in the Upper Athi River Basin as discussed in Chapter 4. In physical sampling, for example, stratified sampling, higher costs are incurred by not following unbiased and representative sampling in the study. It has also proven to be an expensive, cumbersome and time consuming exercise. Thus, the need to develop a statistically sound sampling frame to guide placement of sample points and choice of minimum number of samples is a breakthrough in environmental engineering studies. The development of an LHS extension in ArcView and successful testing in a typical environmental engineering problem is a great achievement. It is the solution to the disadvantages of physical sampling mentioned above.

Thus, some of the applications of LHS extension in ArcView can be summarized as follows:

a) It can be used by statistics and population census department.
b) It can be used by policy makers and administrators as they allocate and reallocate resources.
c) It can be used in telecommunications projects in setting out and monitoring activities.
d) Health care: In addition to the asset management tasks in which GIS is commonly used, the analytical capabilities of GIS aid in sampling can be used in assessing epidemiological effects on population and in carrying out an analysis of outbreak and spread of illnesses and diseases within a community.

e) Retail: Most new out-of-town supermarkets are sited with the aid of a advanced GIS. The GIS is used to store socio-economic data/information and here LHS in a GIS environment is utilized to determine possible customers within the proposed area based on demographics.

5.2 Recommendations

For further success, improvements and wide applications of the LHS extension in ArcView, the following are the three key recommendations.

a) The LHS extension developed in this study only contains Random point sampling and Systematic point sampling features. These two were discussed in detail. It is recommended that the other features in the LHS extension be developed and further discussed in detail with relevant engineering applications cited.

b) LHS extension needs to be tried in freely available software. This would help make the extension more available and more robust in its uses and for wider applications. Use of freely available software in LHS extension also helps in its continual improvements and developments to capture great variability in studies and research works.

c) LHS extension needs to be uploaded in ESRI website www.esri.com. This would make its use available to users online. This would also open great fora for environmental engineers and other researchers to interact and exchange ideas and opportunities for LHS extension. This would help in testing LHS extension robustness and accuracy in its applications.

REFERENCES

Abbaspour K.C., Yang J., Maximov I., Siber R., Bogner K., Mieleitner J., Zobrist J. and Srinivasan R. 2007. *Modelling Hydrology and Water Quality in the Pre-Alpine/Alpine Thur Watershed using SWAT.* Journal of Hydrology, 333:413-430.

Acker R. J. and Brown P. H. 2001. *"The Use of Geographic Information Systems (GIS) in Institutional Research."* Presented at the 41st AIR Forum, Long Beach, California, USA.

Ascough J.C., Baffaut C., Nearing M.A. and Liu B.Y. 1997. *The WEPP Watershed Model: I. Hydrology and Erosion.* Transactions of the American Society of Agricultural Engineers 40(4): 921-933.

Armstrong-Schellenberg J.R.M., Abdulla S., Nathan R., Mukasa O. and Marchant T.J. 2001. *Effect of Large Scale Social Marketing of Insecticide-treated Nets on Child Survival in Rural Tanzania.* Journal of Tropical Medicine 357:1241–1247.

Arnold J.G., Allen P.M. and Bernhardt G. 1993. *A comprehensive Surface-Groundwater Flow Model.* Journal of Hydrology 142: 47-69.

Arnold J.G., Srinivasan R., Muttiah R.S. and Allen P.M. 1998a. *Continental Scale Simulation of the Hydrologic Balance.* Journal of American Water Resources Association. Journal of Hydrology 182: 37-59.

Arnold J.G., Srinivasan R., Muttiah R.S. and Williams J.R. 1998b. *Large Area Hydrologic Modeling and Assessment.* Part I: Model development. Journal of American Water Resources Association, 34 (1): 73-89.

Bartlett J.G., Gonzales R., Besser R.E., Cooper R.J., Hickner J.M. and Hoffman J.R. 2001. *Principles of Appropriate Antibiotic Use for Treatment of Acute Respiratory Tract Infections in Adults*: Background, Specific Aims, and Methods. Ann Intern Med. 134:479-86.

Beven K. and Binley A. 1992. *The Future of Distributed Models - Model Calibration and Uncertainty Prediction*. Hydrological Processes, 6(3): 279-298.

Djokic D. and Ye Z. 1999. *DEM Preprocessing for Efficient Watershed Delineation.* 1999 ESRI User Conference, San Diego, CA.

Favis-Mortlock D.T. 1998. *Validation of Field-scale Soil Erosion Models Using Common Datasets: Modelling Soil Erosion by Water*. Springer-Verlag NATO-ASI Series I-55, Berlin. pp. 89-128.

Flanagan D.C. and Nearing M.A. 1995. *USDA-Water Erosion Prediction Project Hillslope Profile and Watershed Model Documentation.* NSERL Report No. 10, USDA-ARS National Soil Erosion Research Laboratory, West Lafayette, Indiana.

Harrington R. A. 2000. *"Geographical Characterization of Applicants Using the Admitted Student Questionnaire."* Presented at the 40th AIR Forum, Cincinnati, Ohio, May 21–24, 2000.

Hengl T., Heuvelink G.B.M. and Rossiter D.G. 2007. *About Regression-Kriging from Equations to Case Studies.* Computers and Geosciences, 33(10): 1301-1315.

Hutchinson S. and Larry D. 1999. *Inside ArcView GIS.* OnWord Press, USA.

Iman R.L. and Conover W.J. 1981. *Small Sample Sensitivity Analysis Techniques for Computer Models, with an Application to Risk Assessment.* Communications in Statistics Theory and Methods A9, 1749–1874.

Jenson S. and Domingue J. 1988. *Extracting Topographic Structure from Digital Elevation Data for Geographic Information System Analysis.* Photogrammetric Engineering and Remote Sensing, v.54, pp.1593-1600.

Kelsey J.L., Thompson W. D. and Evans A. S. 1986. *Methods in Observational Epidemiology*. New York: Oxford University Press.

Khaemba B.M., Mutani A. and Bett M.K. 1994. *Studies on the Anopheline Mosquitoes Transmitting Malaria in a Newly Developed Highland Urban Area: A Case Study of Moi University and its Environs.* East Africa Medical Journal. 71:159–164.

Kottegoda N.T. and Rosso R. 1997. *Statistics, Probability and Reliability Methods for Civil and Environmental Engineers.* McGraw-Hill College (February 1998), NY 735pp.

Longley P. A., Goodchild M. F., Maguire D. J. and Rhind D. W. 2001. *Geographic Information Systems and Science.* New York: Wiley.

Longley P. A., Goodchild M. F., Maguire D. J. and Rhind D. W. 2005. *Geographic Information System: Principles, Techniques, Management and Applications.* New York, NJ, Wiley.

Mailloux M. R. and Blough D.R. 2000. *"The Who and Where of Adults in Higher Education: Merging Modeling Techniques with GIS to Understand Participation in Credential Courses."* Presented at the 40th AIR Forum, Cincinnati, Ohio, May 21–24.

Manderson L and Aygepong I.A. 1999. *Mosquito Avoidance and Bed Net Use in the Greater Accra Region, Ghana*. Journal of Biosocial Sciences. 31:79–92.

McKay M.D., Beckman R.J. and Conover W.J. 1979. *A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code.* Technometrics 21, 239–245.

Minasny B. and McBratney A.B. 2006. *A Conditioned Latin Hypercube Method for Sampling in the Presence of Ancillary Information.* Computers & Geosciences 32, 1378-1388.

Minasny B. and McBratney A.B. 2007. *Uncertainty Analysis for Pedotransfer Functions.* European Journal of Soil Science 53, 417–430.

Nash J. E. and Sutcliffe J. V. 1970. *River Flow Forecasting through Conceptual Models part I* — A Discussion of Principles. Journal of Hydrology, 10 (3): 282–290.

Nicks A.D. 1985. *Generation of Climate Data*: Proceedings of the Natural Resources Modeling Symposium. USDA-ARS-30. 297-300.

Nielsen D.R. and Wendroth O. 2003. *Spatial and temporal statistics—Sampling Field Soils and their Vegetation.* Catena, Reiskirchen, Germany.

Olivera F., Seann R. and Maidment D. 1998. *HEC-PrePro v. 2.0: An ArcView Pre-Processor for HEC's Hydrologic Modeling System.* 1998 ESRI User Conference, San Diego, CA.

Omuto C.T. 2008. *Assessment of Soil Physical Degradation in Eastern Kenya by Use of a Sequential Soil Testing Protocol.* Agriculture, Economic and Environment. Volume 128, Issue 4, December 2008, Pages 199-211.

Omuto C.T., Minasny B., McBratney A.B. and Biamah E.K. 2006. *Nonlinear Mixed Effects Modelling for Improved Estimation of Water Retention and Infiltration and Parameters.* Journal of Hydrology, 330: 748-758.

Omuto C.T. and Vargas R.R. 2009. Combining Pedometrics, *Remote Sensing and Field Observations for Assessing Soil Loss in Challenging Drylands: A Case Study of Northwestern Somalia.* Land Degradation and Development 20: 101-115 (2009).

Pagano M. and Gauvreau K. 2000. *Principles of Biostatistics.* Pacific Grove, CA: Duxbury.

Pottle L. 2001. *"Geographic Information System (GIS) Applications at a Multi-Site Community College."* Presented at the 41st AIR Forum, Long Beach, Calif., June 3–6.

Renard K.G., Foster G.R., Weesies G.A., McCool D.K. and Yoder D.C. 1997. *Predicting Rainfall Erosion Losses - A Guide to Conservation Planning with the Revised Universal Soil Loss Equation (RUSLE).* U.S. Dept. of Agriculture, Agricultural Handbook 703. 404 p.

Renschler C.S., Engel B.A. and Flanagan D.C. 2000. *Strategies for Implementing a Multi-Scale Assessment Tool for Natural Resource Management: A Geographical Information Science Perspective in Problems, Prospects And Research Needs.* Proceedings of the 4th International Conference on Integrating GIS and Environmental Modeling (GIS/EM4); 2000 Sep 2-8; Banff, Alberta, Canada.

Salant P. and Dillman D. A. 1994. *How to Conduct your Own Survey. John Wiley & Sons.*

Saunders W. 1999. *Preparation of DEMs for Use in Environmental Modeling Analysis.* 1999 ESRI International User Conference, San Diego, CA. USDA Soil Conservation Service (1992). State Soil Geographic Database (STATSGO) Data Users'Guide. Publ. No. 1492, US Government Printing Office, Washington, DC.

Skidmore A.K. 2002. *Environmental Modeling with GIS and Remote Sensing.* Taylor & Francis, NY 268pp.

Stein M. 1987. *Large Sample Properties of Simulations Using Latin Hypercube Sampling.* Technometrics 29:143-151.

Stocking M.A and Murnaghan N. 2001. *A Handbook for the Field Assessment of Land Degradation*; 1853838314; Earthscan Publications Ltd.

Van Griensven A. and Meixner T. 2006. *Methods to Quantify and Identify the Sources of Uncertainty for River Basin Water Quality Models.* Water Science and Technology, 53(1): 51-59.

Webster M. 1985. Webster`s ninth new collegiate dictionary. Meriam - Webster Inc.

Wischmeier W.H. 1976. *Use and Misuse of the Universal Soil Loss Equation.* Journal of Soil and Water Conservation 31 (1): 5-9.

Wischmeier W.H. and Smith D. 1978. *Predicting Rainfall Erosion Losses: A Guide to Conservation Planning.* USDA-ARS Agriculture Handbook N. 537, Washington DC. 58 p.

Wu J. and Zhou Y. 2001. *"Building a University Student Enrollment Database with ArcView GIS: A Step-by-Step Demonstration."* Presented at the 41st AIR Forum, Long Beach, Calif., June 3–6.

APPENDICES

Appendix A: ArcView script for generating histogram

```
' Title:  Creates a histogram for the active theme
' Name:  Create histogram from a raster map
'
' Author: Festus K. Ng'eno
'         University of Nairobi
'         Department of Environmental and Biosystems Engineering
'         P.O. Box 30197-0100
'         Nairobi
'         fkngenoh@yahoo.com
'         (254) 722 989 930

' Date: June 20th 2009

'
' Description:  Generates a histogram for the active theme in the current
' view. A new Chart document is created to display the histogram. A temporary
' file is created to store interval counts and other information used to
' create the histogram. The color scheme used to create the chart will be the
' same as the legend of the active theme, i.e. there is direct correlation
' between the theme classification colors and chart colors.
'
' Should be associated with the Click property of a button on the View
' DocGUI.  An update script should be associated with the control to ensure
' that this script can only be executed when there is at least the user active
' theme.
'
' Requires:  A View with an active theme.  The theme must be classified.
'
' Self:
'
' Returns:


theView=av.GetActiveDoc
theTheme=theView.GetActiveThemes.Get(0)

' Get the compthe usernts of the Legend that will be used to create the chart...
'
theLegend=theTheme.GetLegend
theSymbols=theLegend.GetSymbols
theClasses=theLegend.GetClassifications
theFieldName=theLegend.GetFieldNames.Get(0)

theVTab=theTheme.GetFTab
theField = theVTab.FindField(theFieldName)

' Create the new data file for the histogram...
'
outFName = av.GetProject.MakeFileName( theTheme.GetName, "dbf")
outFName = FileDialog.Put( outFName, "*.dbf", "Output Histogram File" )
if (outFName = Nil) then
  exit
end

newVTab=VTab.MakeNew( outFName, dBASE )
labelf=Field.Make( "Label", #FIELD_CHAR, 20, 0 )
countf=Field.Make( "Count", #FIELD_DECIMAL, 10, 0)
newVTab.AddFields( {labelf, countf} )


' Loop through the classes recording the ranges.
'
countlist = {}
for each c in theClasses
  countlist.Add(0)
end
numClasses = theClasses.Count
```

```
' Loop through the records recording which class they fall in.
'
for each rec in theVTab
  v = theVTab.ReturnValue(theField,rec)
  for each i in 0..(numClasses - 1)
    if (theClasses.Get(i).Contains(v)) then
      countlist.Set(i,countlist.Get(i)+1)
      break
    end
  end
end


' Loop through the classes writing the information to disk.
'
maxcount=0
for each i in 0..(numClasses-1)
  rec=newVTab.AddRecord
  newVTab.SetValue(labelf,rec,theClasses.Get(i).GetLabel)
  newVTab.SetValue(countf,rec,countlist.Get(i))
  maxcount=maxcount max countlist.Get(i)
end


' Create a chart and match the colors to the legend.
'
newChart=Chart.Make(newVTab,{countf})
newChart.SetRecordLabelField(labelf)
for each i in 0..(numClasses-1)
  newChart.GetChartDisplay.SetSeriesColor(i,theSymbols.Get(i).GetColor)
end
newChart.GetTitle.SetName("Histogram of"++theTheme.GetName)
newChart.GetYAxis.SetBoundsUsed(true)
newChart.GetYAxis.SetBoundsMin(0)
newChart.GetYAxis.SetBoundsMax(maxcount)
newChart.GetWin.Open
```

Appendix B: ArcView script for converting histograms into polygons

```
' Title:  Converts histograms to polygons.
'
' Author:    Festus K. Ng'eno
'
'Description: The script  converts histogram into polygons.
'
' Returns: nothing
'prompt the user for the view containing the themes to be edited
theViewList = list.Make
theDocs = av.GetProject.GetDocs
for each theDoc in theDocs
if (theDoc.is(View)) then
theViewList.Add (theDoc)
end
end
theView = msgBox.choiceAsString (theViewList, "Select the view containing the themes", "Select a
View")
if (theView = nil) then
exit
end
 'prompt the user to choose the edit and source themes
theThemesList = theView.GetThemes
theEditTheme = msgBox.ChoiceAsString (theThemesList, "Select the edit theme", "Edit Theme")
if (theEditTheme = nil) then
exit
end
theSame = true
while (theSame)
theSame = false
theSourceTheme = msgBox.ChoiceAsString (theThemesList, "Select the source theme", "Source Theme")
if (theSourceTheme = nil) then
exit
end
if (theEditTheme = theSourceTheme) then
msgBox.error ("The source theme cannot be the same as the edit theme." + NL + "Please choose a
different source theme.", "")
theSame = true
end
end

'find out if the user wants to add fields to the edit theme's attribute table
theEditFTab = theEditTheme.GetFTab
theSourceFTab = theSourceTheme.GetFTab
theEditFieldsList = List.Make
theSourceFieldsList = List.Make
numberOfFields = 0
yesno = msgBox.YesNo ("Do you want to add a new field to the edit table?", "Add Fields?", true)
while (yesno)
theVariable = msgBox.input ("Enter a name for the field that you want to add to the edit theme",
"Edit field", "")
if (theVariable = nil) then
exit
end
theEditFieldsList.Add (theVariable)
theSourceFieldNames = theSourceFTab.GetFields
theVariable = msgBox.ChoiceAsString (theSourceFieldNames, "Select the source field containing the
data you want to add to the " + theVariable.AsString + " field in the edit table", "Source
field")
theFieldType = theVariable.GetType
theFieldWidth = theVariable.GetWidth
theFieldPrecision = theVariable.GetPrecision
theSourceField = {theVariable, theFieldType, theFieldWidth, theFieldPrecision}
if (theVariable = nil) then
exit
end
theSourceFieldsList.Add (theSourceField)
theEditBitmap = theEditFTab.GetSelection
theSourceBitmap = theSourceTheme.GetFTab.GetSelection
theEditBitmap.ClearAll
theSourceBitmap.ClearAll
numberOfFields = numberOfFields + 1
yesno = msgBox.YesNo ("Do you want to add another new field to the edit table?", "Add Fields?",
true)
end

'find out if the user wants to edit existing fields in the attribute table
```

```
yesno = msgBox.YesNo ("Do you want to replace or edit data in existing fields in the edit table?", "Replace data in existing fields?", true)
while (yesno)
theEditFieldNames = theEditFTab.GetFields
theVariable = msgBox.ChoiceAsString (theEditFieldNames, "Select the field that you want to edit", "Edit field")
if (theVariable = nil) then
exit
end
theFieldType = theVariable.GetType
theFieldWidth = theVariable.GetWidth
theFieldPrecision = theVariable.GetPrecision
if (theVariable = nil) then
exit
end
theEditField = {theVariable, theFieldType, theFieldWidth, theFieldPrecision}
theEditFieldsList.Add (theEditField)
theSourceFieldNames = theSourceFTab.GetFields
theVariable = msgBox.ChoiceAsString (theSourceFieldNames, "Select the source field corresponding to the " + theVariable.asString + " field in the edit table", "Source field")
if (theVariable = nil) then
exit
end
theFieldType = theVariable.GetType
theFieldWidth = theVariable.GetWidth
theFieldPrecision = theVariable.GetPrecision
theSourceField = {theVariable, theFieldType, theFieldWidth, theFieldPrecision}
if (theVariable = nil) then
exit
end
theSourceFieldsList.Add (theSourceField)
theEditBitmap = theEditFTab.GetSelection
theSourceBitmap = theSourceTheme.GetFTab.GetSelection
theEditBitmap.ClearAll
theSourceBitmap.ClearAll
numberOfFields = numberOfFields + 1
yesno = msgBox.YesNo ("Do you want to choose another existing field in the edit table?", "Add data to existing fields?", true)
if (yesno = nil) then
exit
end
end

if (numberOfFields = 0) then
exit
end

'Add the new field(s) to the table, or check to see that existing
'fields are of compatible types with the corresponding source fields.
'Also create a final list of fields to be edited called theEditFields.
theEditFields = List.Make
theNewEditFields = List.Make
theCount = theEditFieldsList.Count
theIndexA = 0
theBadTypeList = List.Make

'loop through all the new or existing fields the user has specified
while (theIndexA < theCount)

'check to see if the user has chosen an existing field or if it's a
'new field the user wants to add (if it's a string, it's a new field)
if (theEditFieldsList.Get(theIndexA).is(String)) then

'check to see if the user wants to add fields that already exist. If
'they don't already exist, make them.
if (theEditFTab.FindField (theEditFieldsList.Get(theIndexA).asString) = nil) then
theType = theSourceFieldsList.Get(theIndexA).Get(1)
theWidth = theSourceFieldsList.Get(theIndexA).Get(2)
thePrecision = theSourceFieldsList.Get(theIndexA).Get(3)
theNewField = Field.Make(theEditFieldsList.Get(theIndexA), theType, theWidth, thePrecision)
theEditFields.Add (theNewField)
theNewEditFields.Add (theNewField)
theIndexA = theIndexA + 1

'If they do already exist, add the field names to theEditFields.
else
theEditFields.Add (theEditFieldsList.Get(theIndexA))
theIndexA = theIndexA + 1
end

'check that existing edit fields are the same type as the source field.
```

```
'If not, exit with an error message.
elseif (theEditFieldsList.Get(theIndexA).Get(1) <> theSourceFieldsList.Get(theIndexA).Get(1))
then
theBadTypeList.Add (theEditFieldsList.Get(theIndexA).Get(0))
theBadTypeList.Add (theEditFieldsList.Get(theIndexA).Get(1).asString)
theBadTypeList.Add (theSourceFieldsList.Get(theIndexA).Get(0))
theBadTypeList.Add (theSourceFieldsList.Get(theIndexA).Get(1).asString)
theIndexA = theIndexA + 1
'add them to theEditFields.
else
theEditFields.Add (theEditFieldsList.Get(theIndexA).Get(0))
theIndexA = theIndexA + 1
end
end

'Exit with an error message if there are incompatible field types, after removing the bad edit
fields
theBadTypeListCount = theBadTypeList.count
if (theBadTypeListCount > 0) then
theIndexB = 0
while (theIndexB < theBadTypeListCount)
msgBox.Error ("Incompatible field types:" + NL + "The field " +
theBadTypeList.Get(theIndexB).asString.quote + " = " + theBadTypeList.Get(theIndexB+1) + NL +
"The field " + theBadTypeList.Get(theIndexB + 2).asString.quote + " = " +
theBadTypeList.Get(theIndexB + 3) + NL + "The edit and source fields must be the same type.",
"Error")
theIndexB = theIndexB + 4
end
exit
end

'add theEditFields
theEditFTab.setEditable (true)
if (theEditFields.count > 0) then
theEditFTab.AddFields (theNewEditFields)
end

'Loop through all the records
for each theRowNumber in 0 .. (theEditFTab.GetNumRecords - 1)
theIndex = 0

'Overlay with theme
theShapeField = theEditFTab.FindField ("Shape")
theShape = {theEditFTab.ReturnValue (theShapeField, theRowNumber)}
theEditBitmap.ClearAll
theEditBitmap.Set(theRowNumber)
theDistance = 0
theSourceTheme.SelectByTheme (theEditTheme, #FTAB_RELTYPE_CONTAINSTHECENTEROF, theDistance,
#VTAB_SELTYPE_NEW)
theSourceBitmap = theSourceTheme.GetFTab.GetSelection

'Loop through all the fields to be populated
while (theIndex < numberOfFields)

'if a source table row was selected
if (theSourceBitmap.Count = 1) then

'if the edit table field is blank for the selected row
theField = theEditFTab.FindField (theEditFields.Get(theIndex).asString)

'Populate the edit table fields with the source table values for the selected row
for each theSourceRow in theSourceBitmap
theValue = theSourceTheme.GetFTab.ReturnValue (theSourceFieldsList.Get(theIndex).Get(0),
theSourceRow)
theEditFTab.SetValue (theEditFields.Get(theIndex), theRowNumber, theValue)
theIndex = theIndex + 1
end
elseif (theSourceBitmap.count = 0) then
theIndex = theIndex + 1
elseif (theSourceBitmap.count > 1) then
theIndex = theIndex + 1
msgBox.Error ("More than one source polygon contains the center of one of the edit polygons.
Skipping this polygon.","Error")
end
end
end

'clear the selection and stop editing the table
theEditBitmap.ClearAll
theSourceBitmap.ClearAll
theEditFTab.setEditable (false)
```

Appendix C: ArcView script for random point sampling

```
' Name: Develop an add-on program in Arcview GIS for implementing LHS
'
' Author: Festus K. Ng'eno

' Description: This script creates a set of random points that fall within the polygons within a
selected base theme.
'              The points start getting placed at the selected themes extent origin
'              usually in the lower left of the theme. Points are then created
'              row by row, with the points falling within the polygons in the selected
'              theme being sequentially coded from 1..n.
'              The new points are saved in a shapefile.....

' Requires: null
' Runs: null
' Run by: null
' Self: null
' Returns: null
theView = Av.GetActiveDoc
Viewgr = theView.GetGraphics
distunits = theView.GetDisplay.GetDistanceUnits
mapunits = theView.GetUnits
d = units.GetUnitString(distunits)
ShapeList = {}
GROUP = FALSE     ' do we want to treat the selected graphics/polygons as a single or seperate
polygons.
tmpshp = nil

' see if there are some graphics selected. If so, generate the list of shapes to process....
'
if (theView.GetGraphics.GetSelected.count > 0) then
  numpolys = 0
  numlines = 0
  for each agr in theView.GetGraphics.GetSelected
    if (agr.getshape.getDimension = 1) then
      numlines = numlines + 1
    elseif (agr.getshape.getdimension = 2) then
      numpolys = numPolys + 1
    end
  end
  if ((numpolys > 0) and (numlines > 0)) then
    msgbox.Error("You have mixed types of shapes selected. You must select all Polygons or all
Lines to continue",Script.The.GetName)
    return nil
  end
  if (numpolys > 0) then
    shpType = "polygon"
   else
    shpType = "polyline"
  end
  if (msgbox.yesno("Do you want to use the selected Graphics?",Script.The.GetName,TRUE)) then
    if (theView.GetGraphics.GetSelected.Count > 1) then
      GROUP = msgbox.yesno("Multiple shapes selected. Do you want them to be considered a single,
multi-part shape?",Script.The.GetName,FALSE)
    end
    for each agr in theView.GetGraphics.GetSelected
      if (GROUP) then
        if (tmpshp = nil) then
          tmpshp = agr.getshape
         else
          tmpshp = tmpshp.returnunion(agr.getshape)
        end
       else
       ShapeList.Add(agr.GetShape)
      end
    end
  end
end

if (GROUP) then
  ShapeList.Add(tmpShp)
end
tlist = {}
if (ShapeList.Count = 0) then ' we know that no graphics were selected...
  for each athm in theView.GetThemes
    if ( (athm.Is(FTheme)) and (athm.getftab.getshapeclass.getclassname.lcase <> "point")) then
      tList.add(athm)
    end
  end
  thetheme = msgbox.ListAsString(tlist,"Select Shapefile",script.The.GetName)
  if (thetheme = nil) then
    return nil
  end


  theftab = thetheme.getftab
  shpType = theFtab.GetShapeClass.GetClassName.lcase

  if (theftab.getnumselrecords > 0) then
    theset = theftab.getselection
    pcount = theset.count
   else
    if (msgbox.yesno("No "+shpType+"s selected, do you want to use the entire set of
"+shpType+"s?",Script.The.GetName,TRUE).not) then
     return nil
    end
    theset = theftab
    pcount = theftab.getnumrecords
  end
  if (pcount > 1) then
    GROUP = msgbox.yesno("Multiple "+shpType+"s selected. Do you want them to be considered a
single, multi-part "+shpType+"?",Script.The.GetName,FALSE)
  end
  for each rec in theset
    if (GROUP) then
      if (tmpshp = nil) then
        tmpshp = theftab.returnvalue(theftab.findfield("SHAPE"),rec)
       else
        tmpshp = tmpshp.returnunion(theftab.returnvalue(theftab.findfield("SHAPE"),rec))
      end
     else
      tmpshp = theftab.returnvalue(theftab.findfield("SHAPE"),rec)
      ShapeList.Add(tmpshp)
    end
  end
end
if (GROUP) then
  ShapeList.Add(tmpshp)
end

prompts = {"Number of Sample Points","Point Sample Radius ("+d+")","Sample Spacing (Min =
Radius(2))"}
defaults = {"0","0","0"}
arglist = msgbox.MultiInput("Enter Sampling Parameters","Base
File:"++"PointSample.Shp",prompts,defaults)
if ( (arglist = nil) or (arglist.count < 2) ) then
  return nil
end
numpoints = arglist.get(0)
sampleradius = arglist.get(1)
minspacing = arglist.get(2)

if (Numpoints.IsNumber.Not) then
  msgbox.error("Minimum number of points input needs to be a number!",Script.The.GetName)
  return nil
 else
  numpoints = numpoints.asnumber
end
if (sampleradius.IsNumber.Not) then
  msgbox.error("Point Sample Radius input needs to be a number!",Script.The.GetName)
  return nil
 else
  sampleradius = sampleradius.asnumber
  sampleradius = Units.Convert(sampleradius,distunits,mapunits)
end
if (minspacing.IsNumber.Not) then
  minspacing = sampleradius * 2
 else
  minspacing = minspacing.AsNumber
  if (minspacing < (sampleradius*2)) then
    msgbox.info("Invalid Minimum Spacing - Resetting to (Sample Radius * 2)",Script.The.GetName)
    minspacing = sampleradius * 2
  end
end

' make the output file
pntftabname = FileDialog.Put("plotpnts.shp".AsFilename,"*.shp",Script.The.GetName)
if (pntftabname = nil) then
```

```
  return nil
end
pntftab = Ftab.MakeNew(pntftabname,POINT)
idfield = field.make("PLOTID",#FIELD_LONG,5,0)
xfield = Field.Make("XCoord",#FIELD_FLOAT,7,0)
yfield = Field.Make("YCoord",#FIELD_FLOAT,7,0)
pntftab.AddFields({IDFIELD,xfield,yfield})
pntshpfld = pntftab.findfield("Shape")

if (shpType = "polyline") then
  sampleradius = 0
end
j = 1
bufgrlist = {}
for each tmpshp in ShapeList
  therect = tmpshp.ReturnExtent
  theorigin = therect.ReturnOrigin
  minX = theOrigin.Getx
  MinY = theOrigin.Gety
  MaxX = theRect.GetRight
  MaxY = theRect.GetTop
  lshp = tmpshp.AsPolyLine
  SiteID = 0
  numtries = 0
  While (SiteID < NumPoints)
    numtries = numtries + 1
    if (shpType = "polygon") then
      thepnt = Point.Make(Number.MakeRandom(minx,maxx),Number.MakeRandom(miny,maxy))
    else ' it's a polyline file...
      thepnt = tmpshp.along(Number.MakeRandom(0,100))
    end

    if ((tmpshp.contains(thepnt)) or (tmpshp.Intersects(thepnt)) ) then
      if (thepnt.Distance(lshp) >= sampleradius) then
        thebuff = thepnt.ReturnBuffered(sampleradius)
        selbuff = thepnt.ReturnBuffered(minspacing)
        pntftab.SelectByPolygon(selbuff,#VTAB_SELTYPE_NEW)
        if (pntftab.getnumselrecords = 0) then
          bufgrlist.add(GraphicShape.Make(thebuff))
          newrec = pntftab.addrecord
          pntftab.SetValue(pntshpfld,newrec,thepnt)
          SiteID = SiteID + 1
          pntftab.SetValue(idfield,newrec,SiteId)
          pntftab.SetValue(xfield,newrec,thepnt.getx)
          pntftab.SetValue(yfield,newrec,thepnt.gety)
          av.showmsg("Placed "+SiteID.AsString+" points out of "+numpoints.asstring)
        end
      end
    end
    if (numtries > (numpoints * 1000)) then
      numtries = numtries * j
      j = j + 1
      if ( msgbox.YesNo("Tried "+numtries.asstring+" but was unable to create as many points as
you want ("+SiteID.AsString+"). Do you want to keep trying?",Script.The.GetName,TRUE).not) then
        return nil
      else
        numtries = 0
        continue
      end
    end
  end
end
pntftab.SetEditable(false)
newtheme = theme.make(pntftab.GetSRCName)
themegr = newtheme.getgraphics
theView.AddTheme(newtheme)
for each agr in bufgrlist
  Viewgr.Add(agr)
  themegr.add(agr)
end
newTheme.GetGraphics.SetVisible(newTheme.IsVisible)
newTheme.GetGraphics.Invalidate.
```

Appendix D: ArcView script for converting raster to vector

```
' Title:  Converts polygons in active theme to Point
'
' Author:    Festus K. Ng'eno
'
' Description:  Converts selected polygons to Points to create a new shapefile.
' If no features are currently selected all polygons will be processed.
'

theView    = av.GetActiveDoc
thmThemeIn = theView.GetActiveThemes.Get(0)

' Specify the output shapefile...

  fnDefault = FileName.Make("c:\tmp\").MakeTmp("shape","shp")
  fnOutput  = FileDialog.Put( fnDefault,"*.shp","Output Shape File" )
  if (fnOutput = nil) then exit end
  fnOutput.SetExtension("shp")
  ftbOutput = FTab.MakeNew( fnOutput, POINT )
  ftbOutput.AddFields({Field.Make("ID", #FIELD_LONG, 8, 0)})
  ftbOutput.AddFields({Field.Make("New-ID", #FIELD_LONG, 8, 0)})
  ftbOutput.AddFields({Field.Make("X-Coord", #FIELD_DECIMAL, 18, 5)})
  ftbOutput.AddFields({Field.Make("Y-Coord", #FIELD_DECIMAL, 18, 5)})

' Check if having selection

  if (thmThemeIn.GetFTab.GetSelection.Count > 0) then
      colToProcess = thmThemeIn.GetFTab.GetSelection
      nRecs = colToProcess.Count
    else
      colToProcess = thmThemeIn.GetFTab
      nRecs = colToProcess.GetNumRecords
  end

'Get a List of Fieldnames that can be used
  aFields = {}
   for each f in thmThemeIn.GetFtab.GetFields
    if ( f.IsTypeNumber ) then
       aFields.Add(f)
     end
   end

  thisfield= MsgBox.ListAsString (aFields, "List of Fields
in"++thmThemeIn.AsString, "List")
  MsgBox.Info(" You have selected field ---> " ++thisfield.AsString, "Field")
  ftbOutput.AddFields({Field.Make(thisfield.AsString, #FIELD_DECIMAL, 18, 5)})

  nCount = 0
  nRecsAdded = 0
  newid1 = 0
  thisFtab = thmThemeIn.GetFTab
  fldShapeIn  = thmThemeIn.GetFTab.FindField("shape")
  fldShapeOut = ftbOutput.FindField("shape")
  fldIDOut    = ftbOutput.FindField("id")
  fldnewidOut   = ftbOutput.FindField("New-ID")
  fldselectedOut = ftbOutput.FindField(thisfield.AsString)
  fldselectedIn  = thmThemeIn.GetFTab.FindField(thisfield.AsString)
  fldXOut  = ftbOutput.FindField("X-Coord")
  fldyOut  = ftbOutput.FindField("Y-Coord")

  for each r in colToProcess

    nCount = nCount + 1
     av.SetStatus((nCount / nRecs) * 100)
             shpIn = thmThemeIn.GetFTab.ReturnValue(fldShapeIn,r)
     selectedvalue = thmThemeIn.GetFTab.ReturnValue(fldselectedIn, r)

   shpFld = thmThemeIn.GetFTab.FindField("shape")

  for each rec in thisFTab

    shp = thisFTab.ReturnValue(shpFld, rec)
    id = thisFTab.ReturnValue(fldselectedIn, rec)
      if (id.Is(String).Not) then
          id = id.AsString
      end
  For each part in shp.AsList
    For each p in part
       newid1 = newid1 + 1
       shpNew = Point.Make(p.GetX, p.GetY)
      nRecNew = ftbOutput.AddRecord
  ftbOutput.SetValue(fldShapeOut,nRecNew,shpNew)
  ftbOutput.SetValue(fldIDOut,nRecNew,nCount)
  ftbOutput.SetValue(fldnewidOut,nRecNew,newid1)
  ftbOutput.SetValue(fldXout,nRecNew,p.GetX)
  ftbOutput.SetValue(fldYout,nRecNew,p.GetY)
  ftbOutput.SetValue(fldselectedOut,nRecNew,selectedvalue)
  nRecsAdded = nRecsAdded + 1
   end
  end
 end
end

av.SetStatus(100)

if (nRecsAdded = 0) then
   MsgBox.Info(" Unable to convert shapes ", "Convert Polygon to MultiPoint")
  exit
 else
  MsgBox.Info(nRecsAdded.AsString++"shapes converted.",
    "Convert Polygon to PolyLine")
end

if (MsgBox.YesNo("Add shapefile as theme to a view?",
  "Convert Polygon to PolyLine", true).Not) then
  exit
end

' Create a list of views and allow the user to choose to add theme
  lstViews = {}
   for each d in av.GetProject.GetDocs
    if (d.Is(View)) then
      lstViews.Add( d )
     end
   end
  lstViews.Add("<New View>")

 vweAddTo = MsgBox.ListAsString( lstViews,"Add Theme to:",  "Convert Polygon to
Point" )

' Get the specified view, make the theme, and add it...
 if (vweAddTo <> nil) then
  if (vweAddTo = "<New View>") then
    vweAddTo = View.Make
    vweAddTo.GetWin.Open
  end
  thmNew = FTheme.Make( ftbOutput )
  vweAddTo.AddTheme( thmNew )
  vweAddTo.GetWin.Activate
 end
```

Appendix E: ArcView script for extension building

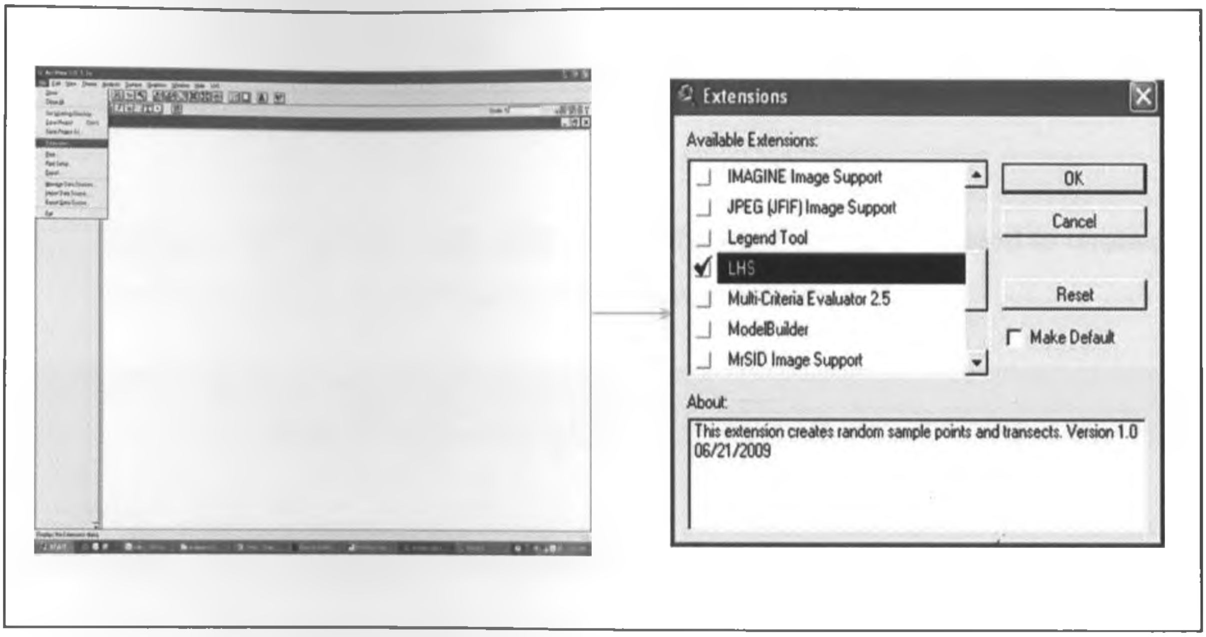
```
/3.0
(Extension.1
	Name:	"Extension Builder Sample"
	FirstRootClassName:	"List"
	Roots:	2
	Roots:	12
	Roots:	13
	Roots:	14
	Version:	30
	About:	"This extension installs 3 scripts to help in building your own extensions."
	InstallScript:	15
	UninstallScript:	16
	ExtVersion:	1
)

(List.2
	Child:	3
	Child:	6
	Child:	9
)

(SEd.3
	Name:	"My Extension MAKE"
	CreationDate:	"Wednesday, May 08, 1996 11:00:05"
	GUIName:	"Script"
	Win:	4
	CSMgr:	5
	Source:	"' Name:  Extension Builder - Source Code\n'\n' Title: Extension Builder - Source
Code\n'\n' Topics:  \n'\n' Description:  \n'\n' Requires:  \n'\n' Self:  \n'\n' Returns:
\n\n\n\n'------------------------------\n 'the Filename of the extension\n '\ntheExtensionFile =
\"$HOME\\sample.avx\"\n\n'------------------------------\n 'The Name of the extension to be
displayed in the\n 'extension dialog\ntheExtensionName = \"Sample Extension\"\n\n'--------------
--------------\n 'The description of the extension to be displayed\n ' in the extension
 dialog\n \ntheDescription = \"This is an example extension\"\n\n'------------------------------
\n 'The version of the extension\ntheVersion = 3.0\n\n'------------------------------\n 'Install
Script\n ' The Script to use to install the extension\n 'NOTE: \"My Extension Install\" is a pre-
perpared script for this\nInScriptName=\"My Extension Install\"\n\n'-----------------------------
-\n 'Uninstall Script\n ' The Script to use to uninstall the extension\n 'NOTE: \"My Extension
Uninstall\" is a pre-perpared script for this\n\nUnScript
Name=\"My Extension Uninstall\"\n\n'------------------------------\n 'Any document Names to
include\n ' This is a list of any documents to include\n ' e.g.
theDocs={\"View1\",\"Layout1\"}\ntheDocs={}\n\n'------------------------------\n 'Controls
(Buttons and normal Tools) to include\n 'Controls {GUI name,ControlType,ScriptName}\n 'e.g.
TheControlList={{\"View\",\"ButtonBar\",\"View.ZoomIn\"},
{\"View\",\"ToolBar\",\"View.Identify\"}}\nTheControllist={}\n\n'------------------------------\n
'Tool menus to include\n ' This is a list of which
tool menus to include.  Tool menus are specified\n ' by giving a GUI name and a script name of
one of the tools in the tool menu\n ' Using this to locate the tool menu ALL other tools inb that
menu and \n ' their scripts will be extracted\n '
theToolMenu={{\"View\",\"View.PointTool\"}}\nTheToolMenuList={}\n\n\n'--------------------------
---\n 'The Menus to include (a list of menu items listing the doc, the top menu, and the script
for the menu item)\n 'The MenuList {Doc name, Main Menu name , menu item Scriptname}\n ' e.
g. The MenuList={{\"View\", \"File\", \"View.Export\"}}\nTheMenuList={}\n\n'---------------------
---------\n 'The scripts to include, not in controls or menus\n 'e.g.
theScripts={\"My.Script\",\"View.export\"}\ntheScripts={}\n\n\n'------------------------------\n
'The Dependencies\ntheDependencies={}\n\n'------------------------------\n'
ATTENTION!!!!!!!!!!!!!!!\n' Do NOT alter the script after these lines, the remainer of this
script\n' uses the lists you provided above to build the extension
object.\n'____________________________
______________________________\n'_______________________________
___\n'_________________________________________________\n'__________
________________________________\n'_________________________________
________\n'___________________________________________________\n'_____________
_____________________________________\n'________________________
___________\n'_____________________
______________________________________\n\n'Create a total list of the needed
scripts\nTheNeeded={}\n\nfor each ControlScript in thecontrolList\n
TheNeeded.add(ConTrolScript.get(2))\nend\n\nFor each MenuScript in theMenuList\n
theNeeded.add(MenuScript.get(2))\nend\n
\ntotalscripts=thescripts.merge(theNeeded)\nTotalScripts.removeduplicates\n
\ntheInstall=av.GetProject.FindScript(InscriptName)\nif (theInstall=NIL) then \n
Msgbox.Error(\"The install Script \"+InScriptName+\" was not found\",\"Script not found\")\n
return(nil)\nend
\n\ntheUninstall=av.GetProject.FindScript(unscriptName)\nif (theUninstall=NIL) then \n
Msgbox.Error(\"The install Script \"+UnScriptName+\" was not found\",\"Script not found\")\n
return(nil)\nend\n\n\n'Create the extension\n' Extension.Make(ExtensionFile, theExtensionName,
InstallScript, UninstallScript, Dependancies_\n\nmyExt =
Extension.Make(theExtensionFile.asFilename, \n theExtensionName,\n
theInstall,\n                                  theUninstall,                                  \n
theDependencies)\n\n\n\nTheDocList={}\nThe
TotalControls={}\nThetotalMenus={}\ntheTotalToolMenus={}\n\n' Process the Document List\nfor each
aDoc    in    TheDocs\n         if    ((av.getproject.findDoc(aDoc)=\"Nil\").NOT)    then\n
TheDocList.Add(av.GetProject.FindDoc(aDoc))\n      else\n                 MsgBox.Warning(\"The  Doc
\"+adoc.asstring+\"  cannot  be  found  in  the  current  project.\",\"Script  Error\")\n
return(nil)\n end\nend\n\n'Add the List of Documents\nMyExt.add(TheDocList)\n\n\n' Process the
control List\n'_____________________________________________\n\n'Controls {GUI,controlType,Scriptnam
e}\n'_____________________________________________\n'_____________________________________\n'__
____________________________________________\n\nfor    each    aControl    in    TheControlList\n
theControlDoc=av.getproject.findGUI(aControl.get(0))\n      if  (theControlDoc=NIL)  then     \n
MsgBox.Warning(\"The GUI \"+aControl.get(0)+\" cannot be found in the current project.\",\"Script
Error\")\n                                   return(nil)\n                                 end\n
thecommand=\"av.getproject.findGUI(\"\"\"+aControl.get(0)+\"\"\").Get\"+acontrol.get(1)\n
'msgbox.info(thecommand,\"\")\n  the
script1=Script.Make(thecommand)\n                              thecontrolset=thescript1.doit(\"\")\n
theFoundControl=TheControlSet.FindbyScript(aControl.get(2))\n     if  (theFoundControl=NIL)  then
\n          MsgBox.Warning(\"No  Control  found  with  the  script  \"+acontrol.get(2)+\"  in
\"+acontrol.get(1),\"Warning\")\n                        return(nil)\n          end\n        if
(theFoundControl.is(ToolMenu)) then \n      MsgBox.Warning(\"The script \"+acontrol.get(2)+\" in
\"+acontrol.get(1)+\" is for a tool menu.\",\"Not Supported\")\n        return(nil)\n     end\n
\n  TheTotalControls
.Add({aControl,thefoundControl,theControlSet.getcontrols.find(thefoundcontrol)})\nend   \n'Add to
Ext\nMyExt.add(TheTotalControls)\n\n'_____________________________________\n\n'The  MenuList
{Doc,                                                                                      Menu,
MenuScript}\n'_____________________________________\n'_____________________________________
____\n'_____________________________________\n\nFor    each    aMenu    in    themenulist\n
mDoc=aMenu.get(0)\n          mMenu=aMenu.get(1)\n          mScript=aMenu.get(2)\n           \n
themDoc=av.getproject.findGUI(mDoc)\n  if (themDoc=NIL) then   \n
MsgBox.Warning(\"The GUI \"+mDoc+\" cannot be found in the current project.\",\"Script Error\")\n
return(nil)\n              end\n          theMbar=av.getproject.findGUI(mDoc).GetMenuBar\n
themenu=theMbar.findbylabel(mMenu)\n  if  (themenu=NiL)  then\n        MsgBox.Warning(\"The menu
named  \"+mMenu+\"  is  not  here.\",\"Script  Error\")\n         return(nil)\n   end\n      \n
themenucontrol=themenu.findbyScript(mScript)\n\n        if     (themenucontrol=NiL)     then\n
MsgBox.Warning(\"The script \"+mScript+\" not found.\",\"Warning\")\n      return(nil)\n  end\n
themenu
itemidx=themenu.getcontrols.find(themenucontrol)\n
thetotalmenus.add({amenu,theMenuControl,theMenuItemIdx})\n\nend\n\nmyext.add(thetotalmenus)\n
\n'_____________________________________\n\n'Process          the          Tool              Menu
List\n'_____________________________________\n'_____________________________________\n'
_____________________________________\n   \nfor    each    aControl    in    TheToolMenuList\n
theControlDoc=av.getproject.findGUI(aControl.get(0))\n      if  (theControlDoc=NIL)  then      \n
MsgBox.Warning(\"The GUI \"+aCon
trol.get(0)+\" cannot be found in the current project.\",\"Script Error\")\n       return(nil)\n
end\n                                          thecontrolset=theControlDoc.getToolbar\n
theFoundControl=TheControlSet.FindbyScript(aControl.get(1))\n     if  (theFoundControl=NIL)  then
\n          MsgBox.Warning(\"No  Control  found  with  the  script  \"+acontrol.get(1)+\"  in
\"+acontrol.get(1),\"Warning\")\n                        return(nil)\n          end\n         if
(theFoundControl.is(ToolMenu).not) then \n      MsgBox.Warning(\"The script \"+aControl.get(1)+\"
is NOT for a tool menu.\",\"User Error\")\n
    return(nil)\n              end                                          \n              \n
TheTotalToolMenus.Add({aControl,thefoundControl,theControlSet.getcontrols.find(thefoundcontrol)})
\nend   \n\nmyext.add(thetotalToolmenus)\n   \n\n'_____________________________________\n\n'
Process                                  the                                          Script
list\n'_____________________________________\n'_____________________________________\n'
_____________________________________\n\nfor    each    ascript    in    TotalScripts\n       if
((av.getproject.findscript(ascript)=\"Null\").NOT)                                         then\n
myExt.Add(av.GetProject.FindScript(a
script))\n   else\n      MsgBox.Warning(\"The script \"+ascript+\" cannot be found in the current
project.\",\"Script      Eror\")\n                                 return(nil)\n      end\nend\n
\nmyExt.SetAbout(theDescription)\nmyExt.SetExtVersion(theVersion)\nmyExt.Commit"
        SearchStr:     "exit"
)

(DocWin.4
        Owner:  3
        X:      90
        Y:      11
        W:      454
        H:      375
)

(CSMgr.5
)

(SEd.6
        Name:   "My Extension Install"
        CreationDate:  "Wednesday, May 08, 1996 11:00:05"
        GUIName:       "Script"
        Win:    7
        CSMgr:  8
        Source: "'DO NOT EDIT!!!\n\nif (av.getproject=nil) then return(nil) end\n\n\ntheDocs =
SELF.get(0)\ntheControlList = SELF.get(1)\ntheMenuList =
SELF.get(2)\ntheToolMenuList=SELF.Get(3)\ntheProject=Av.getproject\n\n\n'Add the Docs\n'\nfor
each adoc in theDocs\n theProject.addDoc(adoc)\nend\n\n'Add the Controls\n'\nfor each
totalControl in theControlList\n   'The Control list\n   acontrol=totalControl.get(0)\n   \n
'The physical control\n   theControl = totalControl.get(1)\n   \n   'The control Index\n
theCindex=totalControl.get(2)\n\n   'Find
 the DocGUI\n   theControlDoc=av.getproject.findGUI(aControl.get(0))\n   if (theControlDoc=NIL)
then   \n      MsgBox.Warning(\"The GUI \"+aControl.get(0)+\" cannot be found in the current
project.\",\"Script Eror\")\n      return(nil)\n    end\n   \n   'This finds the control set
\n    thecommand=\"av.getproject.findGUI(\"\"\"+aControl.get(0)+\"\"\").Get\"+acontrol.get(1)\n
thescript1=Script.Make(thecommand)\n  thecontrolset=thescript1.doit(\"\")\n  \n  'Add the control
to the control set\n  theControlSet.Add(theControl,theCindex
)\nend\n\n\n'Add the menus\nfor each totalcontrol in theMenuList\n  \n   'The Control list\n
acontrol=totalControl.get(0)\n      mDoc=acontrol.get(0)\n      mMenu=acontrol.get(1)\n
mMenuItem=acontrol.get(2)\n\n   'The physical control\n   theControl = totalControl.get(1)\n   \n
'The control Index\n      theCindex=totalControl.get(2)\n\n      'Find the DocGUI\n
theControlDoc=av.getproject.findGUI(aControl.get(0))\n      if (theControlDoc=NIL) then    \n
MsgBox.Warning(\"The GUI \"+aControl.get(0)+\" cannot be found in the c
urrent project.\",\"Script Eror\")\n      return(nil)\n      end\n      \n
theMbar=av.getproject.findGUI(mDoc).GetMenuBar\n    themenu=theMbar.findbylabel(mMenu)\n    if
(themenu=NiL)  then\n      themenu=menu.make\n      themenu.setlabel(mMenu)\n
theMbar.add(themenu,999)\n  end\n  \n  themenu.add(thecontrol, theCindex)\nend\n   \n   \n'Add
the Tool Menus\n\nfor each totalControl in theToolMenuList\n    'The Control list\n
acontrol=totalControl.get(0)\n     \n     'The physical control\n     theControl =
totalControl.get(1)\n  \n  'The con
trol Index\n      theCindex=totalControl.get(2)\n\n      'Find the DocGUI\n
theControlDoc=av.getproject.findGUI(aControl.get(0))\n      if (theControlDoc=NIL) then    \n
MsgBox.Warning(\"The GUI \"+aControl.get(0)+\" cannot be found in the current project.\",\"Script
Eror\")\n      return(nil)\n      end\n     \n     'This finds the control set    \n
thecommand=\"av.getproject.findGUI(\"\"\"+aControl.get(0)+\"\"\").Get\"+acontrol.get(1)\n
thescript1=Script.Make(thecommand)\n
thecontrolset=av.getproject.findGUI(aControl.get(0)).GetToolBa
r\n      \n      'Add the control to the control set\n
theControlSet.Add(theControl,theCindex)\nend\n\n\nav.getproject.setmodified(true)\n\n\n'And the
scripts add themselves\n"
        SearchStr:     "exit"
)
(DocWin.7
        Owner:  6
        X:      -3
        Y:      2
        W:      582
        H:      383
)

(CSMgr.8
)

(SEd.9
        Name:   "My Extension Uninstall"
        CreationDate:  "Wednesday, May 08, 1996 11:00:05"
        GUIName:       "Script"
        Win:    10
        CSMgr:  11
        Source: "'DO NOT EDIT!!!\n\n'The SELF is the Extension\n\ntheDocs =
SELF.get(0)\ntheControlList = SELF.get(1)\ntheMenuList =
SELF.get(2)\ntheToolMenuList=SELF.get(3)\ntheProject=Av.getproject\n\n\n'Add the Docs\n'\nfor
each adoc in theDocs\n If (theProject.finddoc(adoc.getname)<>NIL) then \n
theAnswer=msgbox.yesno(\"Remove the Document \"+adoc.getname+\"?\",\"Remove Document?\",TRUE)\n
if (theAnswer=TRUE) then theProject.RemoveDoc(adoc) end\n end\nend\n\n'Removethe Controls\n'\nfor
each totalControl in theControlList\n   'Get the control list
 from the Ext\n   acontrol=totalControl.get(0)\n   \n   'Get the physical Control\n   theControl
= totalControl.get(1)\n   \n   'Get the Controls Index\n   theCindex=totalControl.get(2)\n\n
'Find the DocGUI for the Control\n    theControlDoc=av.getproject.findGUI(aControl.get(0))\n
if (theControlDoc=NIL) then   \n      MsgBox.Warning(\"The GUI \"+aControl.get(0)+\" cannot be
found in the current project.\",\"Script Eror\")\n      return(nil)\n    end\n   \n   'This
sequence finds the appropiate control set\n   thecommand=
\"av.getproject.findGUI(\"\"\"+aControl.get(0)+\"\"\").Get\"+acontrol.get(1)\n
thescript1=Script.Make(thecommand)\n    thecontrolset=thescript1.doit(\"\")\n\n    'See if the
control is in the set , if so remove it\n   if (theControlSet.GetControls.find(theControl)<>NIL)
then\n      theControlSet.remove(theControl)\n      if (thecontrol = \"ToolBar\") then\n
theControlSet.selectdefault\n      end\n      end\nend\n\n\n'Remove the Menus\n'\nfor each
totalcontrol in theMenuList\n  \n   'The Control list\n   acontrol=totalControl.get(0
)\n      mDoc=acontrol.get(0)\n      mMenu=acontrol.get(1)\n      mMenuItem=acontrol.get(2)\n\n
'The physical control\n    theControl = totalControl.get(1)\n    \n    'The control Index\n
theCindex=totalControl.get(2)\n\n      'Find the DocGUI\n
theControlDoc=av.getproject.findGUI(aControl.get(0))\n      if (theControlDoc=NIL) then    \n
MsgBox.Warning(\"The GUI \"+aControl.get(0)+\" cannot be found in the current project.\",\"Script
Eror\")\n      return(nil)\n    end\n   \n  theMbar=av.getproject.findGUI(mDoc).GetMenuBar\n
```

```
 themenu=theMbar.findbylabel(mMenu)\n  if (themenu=NiL) then\n       MsgBox.Warning(\"The menu
named  \"+mMenu+\"  is  not  here.\",\"Script  Eror\")\n        'return(nil)\n   else\n   \n
thething=themenu.getcontrols.find(thecontrol)\n            if   (thething<>NIL)   then   \n
themenu.remove(thecontrol) \n   end\n   'msgbox.info(themenu.GetControls.count.asstring,\"\")\n
if (themenu.GetControls.count<1) then\n        theMbar.remove(themenu)\n     end\n   end\nend\n
\nfor each totalControl in theToolMenuList\n  'Get the control list from the
Ext\n   acontrol=totalControl.get(0)\n    \n   'Get the physical Control\n   theControl =
totalControl.get(1)\n   \n   'Get the Controls Index\n   theCindex=totalControl.get(2)\n\n
'Find  the  DocGUI  for  the  Control\n   theControlDoc=av.getproject.findGUI(aControl.get(0))\n
if (theControlDoc=NIL) then   \n      MsgBox.Warning(\"The GUI \"+aControl.get(0)+\" cannot be
found in the current project.\",\"Script Eror\")\n       return(nil)\n     end\n   \n   'This
sequence finds the appropiate control set\n  thecontrolset=av.getp
roject.findGUI(aControl.get(0)).GetToolBar\n\n \n   'See if the control is in the set , if so
remove   it\n          if   (theControlSet.GetControls.find(theControl)<>NIL)   then\n   \n
theControlSet.remove(theControl)\n     theControlSet.selectdefault\n   end\n\nend\n   \n\n'And
the scripts delete themselves\n\n\nav.getproject.setmodified(true)\n"
       SearchStr:    "exit"
)

(DocWin.10
       Owner:  9
       Y:      8
       W:      574
       H:      551
)

(CSMgr.11
)

(List.12
)

(List.13
)

(List.14
)
(Script.15
       Name:  "Ext-in"
       SourceCode:   "'DO   NOT   EDIT!!!\n\nif   (av.getproject=nil)   then   return   nil
end\n\n\ntheDocs    =    SELF.get(0)\ntheControlList    =    SELF.get(1)\ntheMenuList    =
SELF.get(2)\ntheToolMenuList=SELF.Get(3)\ntheProject=Av.getproject\n\n\n'Add  the  Docs\n'\nfor
each  adoc  in  theDocs\n  theProject.addDoc(adoc)\nend\n\n'Add  the  Controls\n'\nfor  each
totalControl in theControlList\n   'The Control list\n   acontrol=totalControl.get(0)\n   \n
'The  physical  control\n   theControl  =  totalControl.get(1)\n   \n   'The  control  Index\n
theCindex=totalControl.get(2)\n\n  'Find
the DocGUI\n  theControlDoc=av.getproject.findGUI(aControl.get(0))\n    if (theControlDoc=NIL)
then   \n       MsgBox.Warning(\"The GUI \"+aControl.get(0)+\" cannot be found in the current
project.\",\"Script Eror\")\n      return nil\n     end\n   \n   'This finds the control set  \n
thecommand=\"av.getproject.findGUI(\"\"\"+aControl.get(0)+\"\"\").Get\"+acontrol.get(1)\n
thescript1=Script.Make(thecommand)\n  thecontrolset=thescript1.doit(\"\")\n  \n  'Add the control
to the control set\n  theControlSet.Add(theControl,theCindex)\n
end\n\n\n'Add  the  menus\nfor  each  totalcontrol  in  theMenuList\n   \n    'The  Control  list\n
acontrol=totalControl.get(0)\n         mDoc=acontrol.get(0)\n          mMenu=acontrol.get(1)\n
mMenuItem=acontrol.get(2)\n\n   'The physical control\n   theControl = totalControl.get(1)\n   \n
'The  control  Index\n   theCindex=totalControl.get(2)\n'msgbox.info(thecindex.asstring,\"\")\n\n
'Find  the  DocGUI\n      theControlDoc=av.getproject.findGUI(aControl.get(0))\n          if
(theControlDoc=NIL) then  \n      MsgBox.Warning(\"The GUI \"+aControl
.get(0)+\" cannot be found in the current project.\",\"Script Eror\")\n       return nil\n
end\n               \n              theMbar=av.getproject.findGUI(mDoc).GetMenuBar\n
themenu=theMbar.findbylabel(mMenu)\n   if  (themenu=NiL)  then\n          themenu=menu.make\n
themenu.setlabel(mMenu)\n      theMbar.add(themenu,999)\n  end\n  \n  themenu.add(thecontrol,
theCindex)\nend\n   \n   \n'Add the Tool Menus\n\nfor each totalControl in theToolMenuList\n
'The Control list\n  acontrol=totalControl.get(0)\n  \n  'The physical control\n  theControl
= to
talControl.get(1)\n  \n  'The control Index\n  theCindex=totalControl.get(2)\n\n  'Find the
DocGUI\n  theControlDoc=av.getproject.findGUI(aControl.get(0))\n    if (theControlDoc=NIL) then
\n       MsgBox.Warning(\"The  GUI  \"+aControl.get(0)+\"  cannot  be  found  in  the  current
project.\",\"Script Eror\")\n      return nil\n     end\n   \n  'This finds the control set  \n
thecommand=\"av.getproject.findGUI(\"\"\"+aControl.get(0)+\"\"\").Get\"+acontrol.get(1)\n
thescript1=Script.Make(thecommand)\n  thecontrolset=av.getproject.fi
ndGUI(aControl.get(0)).GetToolBar\n     \n     'Add   the   control   to   the   control   set\n
theControlSet.Add(theControl,theCindex)\nend\n\n\nav.getproject.setmodified(true)\n\n\n'And   the
scripts  add  themselves\n\nMsgbox.Report(\"Three  script  documents  have  been  added  to  your
project.\"++\n\"These   scripts   provide   a   simple   entry   into   building   your   own
extensions.\"++\n\"The \"\"My extension MAKE\"\" script is what you should work with as the
\"++\n\"others depend on that script.\",\"Extension Builder\")\n\nSELF.Unload\nreturn(nil)\n"
)
(Script.16
       Name:  "Ext-UN"
       SourceCode:    "'DO NOT EDIT!!!\n\n'The SELF is the Extension\n\nreturn (nil)\n\n")
```

Appendix F: Uploading LHS extension in ArcView

In order to use LHS extension in sampling for example, the following procedure needs to be followed:

i. Load the extension by clicking on **File --> Extensions...**, scrolling down through the list of available extensions, and then clicking on the checkbox next to the extension called "***LHS.avx***".



Extensions

Available Extensions:

☐ IMAGINE Image Support
☐ JPEG (JFIF) Image Support
☐ Legend Tool
☑ LHS
☐ Multi-Criteria Evaluator 2.5
☐ ModelBuilder
☐ MrSID Image Support

OK

Cancel

Reset

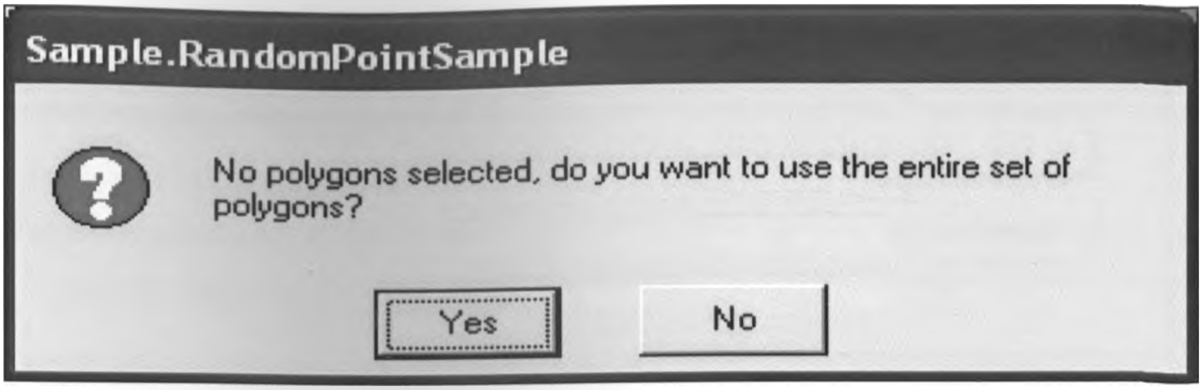
☐ Make Default

About:

This extension creates random sample points and transects. Version 1.0 06/21/2009

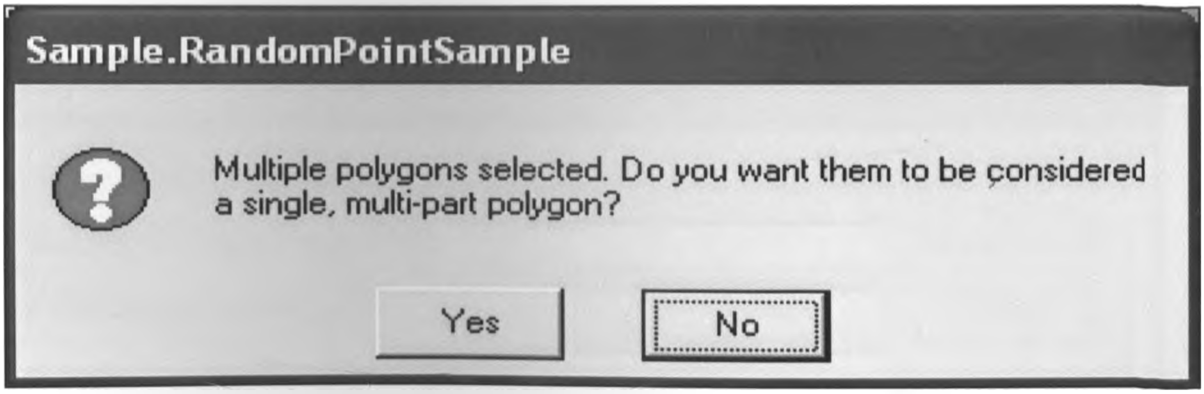
ii. From the LHS extension pull down menu, choose Random Point Sample option as shown below.

iii. The researcher will be prompted to select a shapefile e.g. boundary shapefile for this study as shown below.


Sample.RandomPointSample

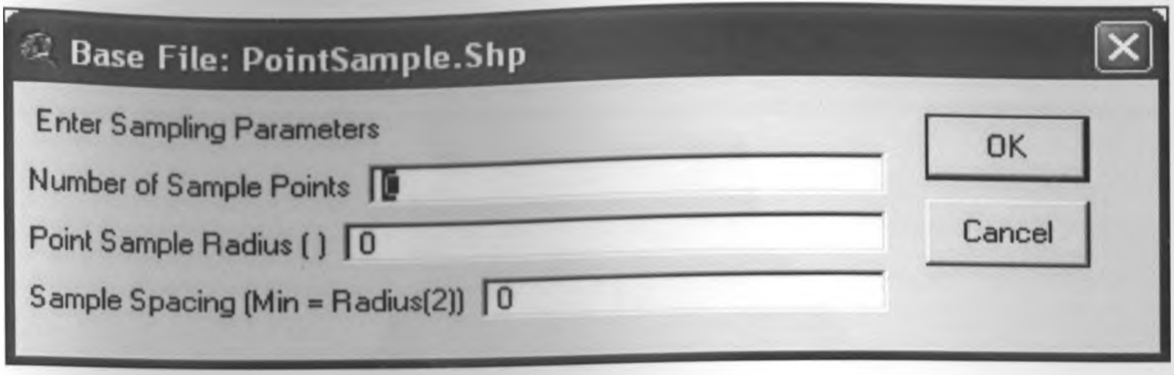
No polygons selected, do you want to use the entire set of polygons?

iv. After selecting the shapefile click ok and the user will be prompted to choose the number of polygons as shown in the following figures.


Sample.RandomPointSample

Multiple polygons selected. Do you want them to be considered a single, multi-part polygon?

v. After choosing the number of polygons from the entire shapefile, the user will then be prompted to choose the number of samples in the study area.



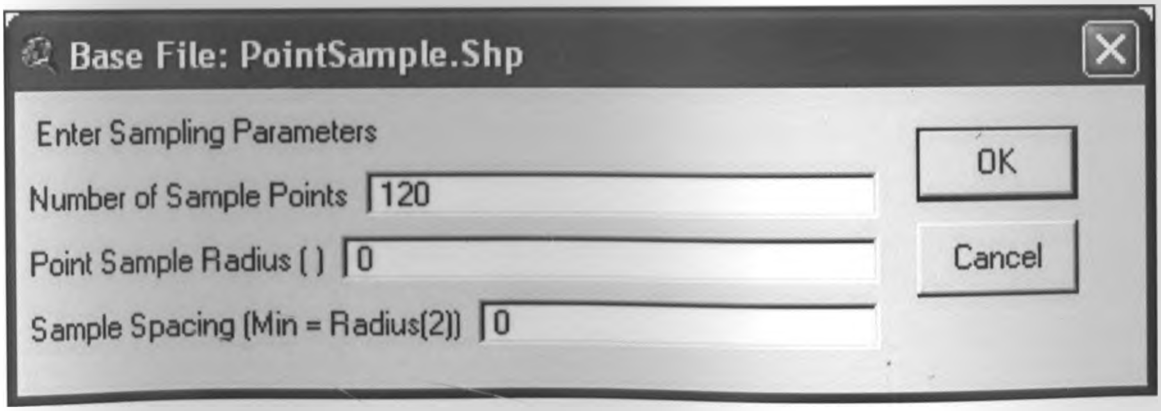
Enter Sampling Parameters

Number of Sample Points

Point Sample Radius ( ) 0

Sample Spacing (Min = Radius(2)) 0

vi. 120 samples were chosen for this study as shown below.



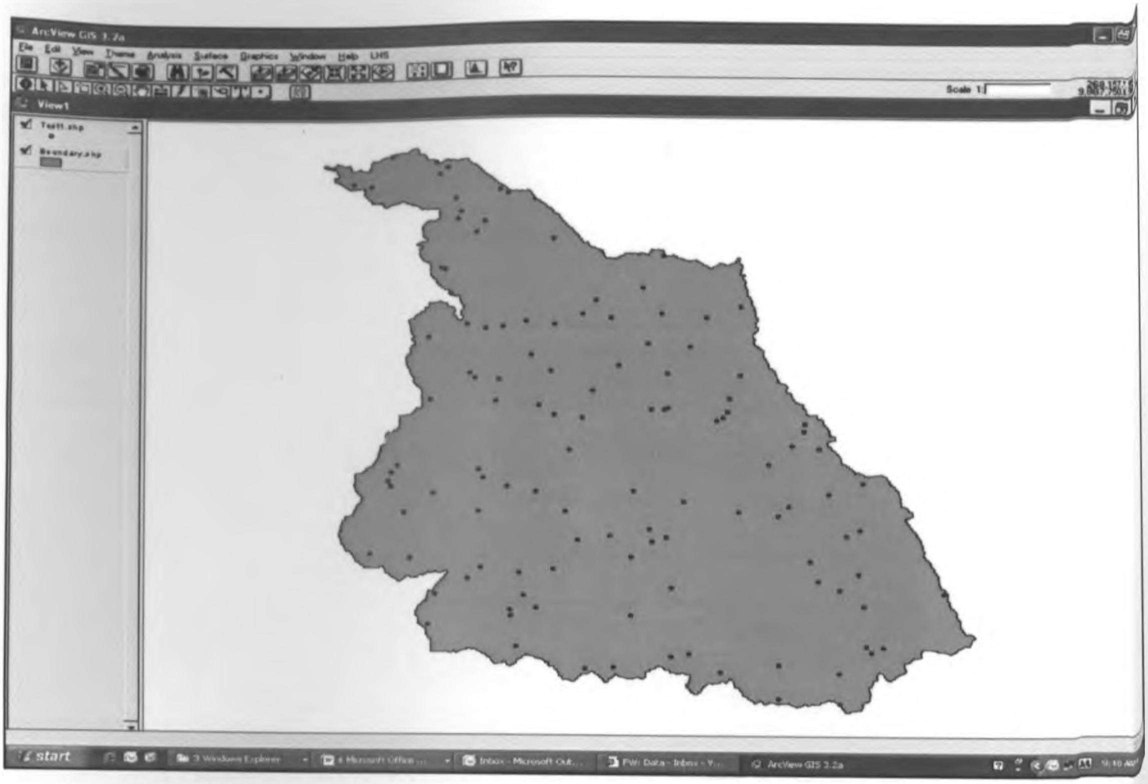
Enter Sampling Parameters

Number of Sample Points 120

Point Sample Radius ( ) 0

Sample Spacing (Min = Radius(2)) 0

vii. The samples will then be distributed in the sampling area as shown below.


ArcView GIS 3.2a
File
Edit
View
Theme
Analysis
Surface
Graphics
Window
Help
LHS
Scale 1:
View1
Test1.shp
Boundary.shp
start
3 Windows Explorer
4 Microsoft Office ...
Inbox - Microsoft Out...
FW: Data - Inbox - Y...
ArcView GIS 3.2a
9:10 AM